
电商物流网络包裹应急调运与结构优化

摘 要

电商物流网络由物流场地（接货仓、分拣中心、营业部等）和物流场地之间的运输线路组成。受节假日和促销活动的影响，电商用户的下单量会发生显著波动，而受疫情、地震等突发事件影响，部分的物流场地可能会被动或主动地临时或永久停用，这时则需要将本应由停用场其处理的包裹紧急分流到其他物流场地。场地的最大运输能力、因素均会影响到各条线路运输的包裹数量以及各个物流场地处理的包裹数量分配。

针对问题一，通过 CRITIC 客观赋权法得到 ARIMA-LSTM 时间序列组合模型函数的关系权重，使用 ARIMA-LSTM 时间序列组合模型分别对不同表格的数据进行预测得到未来 2023-01-01 至 2023-01-31 期间每条线路每天的货量预测值。

针对问题二，首先构建一个物流路线配送能力系数方程，将系数类比为最优路径规划问题中的路径，使用 MDS 算法构建距离矩阵得到所有物流点的功能坐标，将 DC5 关停当天预测货物运输到 DC5 的上一级物流点作为起点，使用多目标蚁群算法进行最优规划，加入启发因子使得物流分配在负荷内分配的条件下物流点变化路线的数量最少。得到 DC5 关闭后的最优路径规划。由于第二问要求不能新增路线，因此需要限制蚂蚁的行动路线以得到新的最优路径规划方案。

针对问题三，构建动态多目标决策流转方案，根据功能坐标，使用蚁群算法进行关停 DC9 的最优路线规划，加入启发因子使得物流分配在负荷内分配的条件下物流点变化路线的数量最少。得到 DC9 关闭后的最优路径规划。分析是否有新增的线路、因关闭 DC9 而导致货量变化的线路数以及各条线路的工作负荷尽可能平衡。提供因 DC9 关停而导致变化的货量线路数量、网络负荷情况和每天的线路增减情况，无法正常流转时另外再提供因 DC9 关停而导致无法正常流转的货物量。

针对问题四，使用中心性度量分析评估附件 1 中不同物流场所和线路的重要性，根据问题一的预测结果探讨新增物流场所和线路对于改善网络性能的影响并确定新的物流场所、线路及处理和运输能力设置。最后构建网络效率以及网络稳定性组合函数检测预测模型的鲁棒性。

关键词：ARIMA-LSTM 时间序列预测 MDS 算法 蚁群算法 中心性度量

目录

目录

一、问题重述	1
1.1 问题背景	1
1.2 问题重述	1
二、问题分析	2
三、模型假设	3
四、符号说明	3
五、数据预处理	4
六、模型的建立与求解	5
6.1 问题一模型的建立与求解	5
6.1.1 时间序列 ARIMA 模型的构建与求解	5
6.1.2 时间序列 LSTM 模型的构建与求解	12
6.1.3 时间序列 ARIMA 与 LSTM 混合模型的构建与求解	12
6.2 问题二模型的建立与求解	15
6.2.1 物流路线配送能力函数的建立	15
6.2.2 MDS 算法的建立与功能坐标的求解	15
6.2.3 优化函数的构建	17
6.3 问题三模型的建立与求解	23
6.3.1 多目标动态函数的建立	23
6.4 问题四模型的建立与求解	26
6.4.1 网络的不同物流场地及线路的重要性评价	26
6.4.2 新物流网络的建立	27
6.4.3 鲁棒性检验	28
七、模型的评价	29
参考文献	30

一、问题重述

1.1 问题背景

随着电子商务的快速发展，物流网络配送成为了电商运营中的核心问题。尤其是2020年新冠疫情以来，电商用户下单量不稳定波动的现象越来越多，因此，建立高效稳定的电商物流网络配送系统来服务电商订单已经成为各大电商平台和公司必须面对的问题。物流企业需要通过提高信息技术应用、利用智能化装备等手段优化物流网络布局，提升物流方案的效率。相比传统物流模式，电商物流需要建立具有覆盖范围广、服务质量齐全、费用透明等特点的物流网点体系，以满足电商行业日益增长的需求。为此，各大电商公司需要建立完善的仓储物流设施并掌握自主配送能力，整合多元化配送资源、保障商品质量符合法规标准等以实现高效便捷、安全可靠的物流体验。

电商物流网络配送的核心价值在于能够向消费者提供快捷、准确和低成本的送货服务。通过合理的物流运作和高效的系统管理实现一体化物流管理，使得商品能够顺利地从仓库出发，经过处理、打包、揽收、运输等环节，最终准时地送达消费者手中。同时，电商物流网络配送中涉及到的各个层面也十分繁杂。在物流运输中，不仅需要面对天气等客观因素带来的影响，还需要考虑订单量、配送距离、客户需求等多种因素。通过基于对未来物流场地线路货量的预测结果和分析各个物流场地的处理能力及线路的运输能力，设计物流网络调整方案，将会大大降低物流场地停用对物流网络的影响，保障物流网络的正常运行。

1.2 问题重述

问题一需要创建一个预测模型，用于预测从2023年1月1日到2023年1月31日之间每条路线每天的货物量，并给出线路DC14→DC10、DC20→DC35、DC25→DC62的预测结果。

问题二需要在物流场地DC5将于2023-01-01开始关停的前提设计一个数学模型，目标是使所有包裹尽可能正常流转，关停前后货量变化的线路数量最小，并且保持各条线路的工作负载均衡。如果部分日期和份额有包裹没有正常流转则调整方案使得2023-01-01至2023-01-31期间未能正常流转的包裹日累计总量尽可能少。给出因DC5关停而导致货量发生变化的线路数、未能正常流转的货量情况以及网络的负荷情况。

问题三将关停的物流场地改改为DC9，假设新开线路的运输能力上线为已有线路运输能力的最大值，不允许新增物流场地但可以动态调整物流网络结构（每日均可调整），通过改变不同线路货量分配值、关闭或新开线路的方式将DC9相关线路的货量分配给其他线路。要求使所有包裹尽可能正常流转，尽可能减少因关闭DC9而导致货量变化的线路数，并保持各条线路的工作负荷尽可能平衡。如果存在日期未满足要求的情况，尽可能减少2023年1月1日至1月31日期间未能正常流转的包裹日累计总量。提供因DC9

关停而导致变化的货量线路数量、网络负荷情况和每天的线路增减情况，无法正常流转时另外再提供因 DC9 关停而导致无法正常流转的货物量。

问题四需要分析评估附件 1 中不同物流场所和线路的重要性，探讨新增物流场所和线路对于改善网络性能的影响并确定新的物流场所、线路及处理和运输能力设置。考虑所建网络的鲁棒性。

二、问题分析

问题一，首先将 81 个场地的 1049 条线路中场地 1→场地 2 相同场地相同方向不同日期及货量的线路分别存入表格中。通过 **CRITIC 客观赋权法**得到 **ARIMA-LSTM 时间序列组合模型**函数的关系权重为 0.5 和 0.5，使用 **ARIMA-LSTM 时间序列组合模型**分别对不同表格的数据进行预测得到未来 2023-01-01 至 2023-01-31 期间每条线路每天的货量预测值，将线路 DC14→DC10、DC20→DC35、DC25→DC62 的预测结果进行了展示。

问题二，首先计算出 1049 条路线的平均载荷以及最大载荷量，以及所有场所点的出货最大值作为处理能力，然后构建了一个物流路线配送能力系数方程，将系数类比为最优路径规划问题中的路径，使用 **MDS 算法**构建出一个功能坐标，坐标越接近的点之间的路线配送能力越强，在运输物流时被优先选择。随后构建**多目标变异蚁群算法**，其目标函数为：1、DC5 关停前后货量发生变化的线路尽可能少；2、DC5 相关线路的货量分配到其他线路使所有包裹尽可能正常流转；3、所有线路上的货物数量尽可能均衡。约束函数：1、路线货物数量不能超过路线载荷；2、保持各条线路的工作负荷尽可能均衡。3、DC5 相关线路上的货物数量分配到其他线路上的总量等于 DC5 相关线路上的货物。本问要求不能新增路线，因此需要限制蚂蚁的行动路线以得到新的最优路径规划方案，由历史数据拟合出最优参数后，依据问题一的结果求解最优的运货量分配组合。

问题三，在问题 2 的基础上，将关停的物流场所改为 DC9，第三问允许对物流网络结构进行动态调整，开启或关闭新路线。使用蚁群算法时不需要限制蚂蚁的行动路线，**构建动态多目标决策流转方案**，在 MD9 周围 0.05 范围内随机生成 2 到 3 个新路线，多次调用多目标规划的变异蚁群算法模型，为减少训练量，我们为每一次迭代都设置了 31 天的运转周期，将问题一的结果作为物流需求，多次迭代后，得到使得目标函数最优的运货量分配组合以及新路线。

问题四，使用分别使用**中心性度量**分析评估附件 1 中不同物流场所和线路的重要性，得到物流点 10，物流点 14，物流点 5 等物流点为主要重要的物流点。通过分析，选取重要性程度前五的坐标作为主要物流点，根据问题二构建的功能坐标求解出新物流点的最优坐标位置。根据问题一的预测结果对路线运输量进行动态调整探讨新增物流场所和线路对于改善网络性能的影响并确定新的物流场所、线路及处理和运输能力设置。最后构建网络效率以及网络稳定性组合函数，通过对**优化前后物流网络进行随机攻击与蓄意**

攻击两种攻击方式检测预测模型的鲁棒性，得到结论优化后的物流网络模型鲁棒性大大增强。

三、模型假设

- 1. 假设剔除异常值后所有数据都是真实没有误差的。
- 2. 假设所有物流场所之间在没有给定条件的情况下都能自由流通，不存在人为干扰。
- 3. 假设所有从一个物流场地到另一个物流场地的直接路线的路况和运送速度相同。
- 4. 假设不同路线和物流场所运输的货物均相同。
- 5. 假设每个物流场地的处理能力和每条线路的运输能力上限均为其历史货量最大值。
- 6. 假设已有的 2 年货量数据中的极端数据是最优值，可以作为路径长度的参考。
- 7. 假设物流路线的货物运输能力及物流场所的货物处理能力上限在不同日期是一样的。
- 8. 假设新开线路的运输能力上限为已有线路运输能力的最大值。

四、符号说明

符号	说明
Z_t	差分后的数据
y_i	原始的物流路线负载值
L	模型的似然函数
N	模型中未知参数的个数
S_t	差分平稳序列
Z_t	随机误差
p	自回归系数
q	移动平均数
y	物流路线负载原始时间序列
f_l	LSTM 模型代表的函数关系
n	窗口大小
W_j	指标权重
p_{ij}	指标率
D_{ij}	物流路线配送能力系数
S_{ij}	物流路线稳定系数
P_{ij}	物流承压能力系数
A_{ij}	物流路线平均载货量

M_{ij}	物流路线最大载货量
O_{ij}	物流点最大出货量
d_{ij}	原始空间中数据点的距离矩阵
B_{ij}	内积矩阵
A_{ij}	特征值矩阵
V_{ij}	特征向量矩阵
$\tau_{ij}^{(t)}$	t 时刻物流点 i 和 j 之间的剩余信息素
η_{ij}	启发性因子
allowed _k	蚂蚁 k 下一步允许选择的物流点
α	信息启发因子
β	期望启发因子
$1 - \rho$	信息消逝程度
ρ	信息素的挥发系数
$\Delta \tau_{ij}^k$	第 k 只蚂蚁在本次循环中留在路径 ij 上的信息量
$\Delta \tau_{ij}$	m 只蚂蚁在本次循环中留在路径 ij 上的信息量
L_k	本次循环中第 k 只蚂蚁所走过的路径的总长度
Q	信息素强度的一个常数
y_t	物流路线负载值

注: $i = 1, 2, \dots, 81; j = 1, 2, \dots, 81; i \neq j$

五、数据预处理

1. 对无效数据剔除。
2. 针对表单的数据赋值量化。
3. 对同一条线路中不同日期的货量取平均值作为此线路的一般需求物流量以简化计算。
4. 统计所有物流场所不同日期的出货量与进货量，将所有日期里出货量最大的日期的货量值作为该物流场所处理能力上限，将所有日期里进货量最大的日期的货量值作为该物流场所处理能力上限。
5. 统计不同日期下同一个物流场所对所有物流路线的进出货量，将所有日期里出货量最大的路线的值作为该路线出货配送能力上限，将所有日期里进货量最大的值作为该路线进货配送能力上限。

6. 将数据重新处理读入 excel 中使得数据转化为二维数据，其中行标签为日期，列标签为不同路线。对不同线路在不同日期下的值先剔除异常值再做平均处理得到各线路的日平均货量。

六、模型的建立与求解

6.1 问题一模型的建立与求解

6.1.1 时间序列 ARIMA 模型的构建与求解

根据问题一，需要预测出 2023 年 1 月 1 日-2023 年 1 月 31 日的货量，自回归综合移动平均模型 (ARIMA) 是时间序列预测分析方法之一，可以对时间序列数据的未来走势进行预测，用于检测和分析时间序列数据的周期性、趋势性和季节性等特征。ARIMA 模型建立过程主要包括数据平稳性检测、数据差分处理、模型定阶、模型检验以及模型结果预测，ARIMA 模型构建流程如图所示。

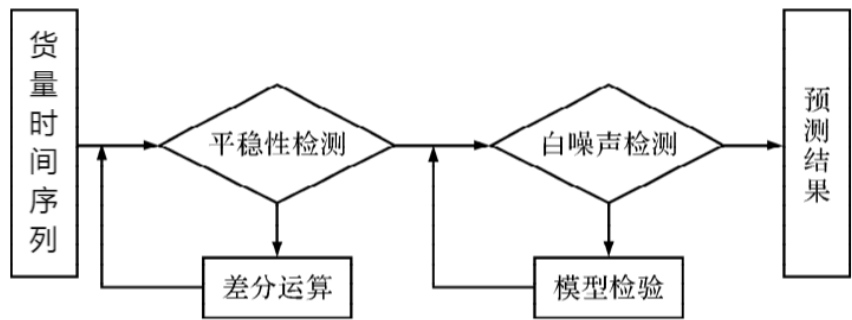


图 6.1.1 ARIMA 模型的构建

下面是 ARIMA 时间序列预测的具体步骤。

首先是数据平稳性检测。时间序列平稳性检测最常用的就是观察法。观察法通过货量时间序列变化趋势来判断是否平稳，判断依据是如果时间序列值只在某个常数附近随机变化并且没有出现周期规律的趋势，数据平稳；如果时间序列值存在单增或单减趋势，数据非平稳。除了观察时间序列自身变化趋势判定是否平稳外，还可以通过观察时间序列的观察 ACF 自相关图和偏自相关图所表现的特征来判定数据平稳性。通过计算自相关函数和偏自相关函数来画图，其中通过自相关函数如式 (1) 可以获取自相关数如式 (2)。

$$\gamma(t,s) = E(y_i - \bar{y})(y_{t+k} - \bar{y}) \tag{6.1.1}$$

$$\hat{P}_k = \frac{\gamma(t,s)}{\sum_{t=1}^n (y_i - \bar{y})^2} \tag{6.1.2}$$

其中， y_t 为物流路线负载值， \bar{y} 表示是在 n 时间段内货量平均数； E 表示期望。偏自相关函数如式 (3) 所示。

$$\varphi = \frac{D_k}{D} \tag{6.1.3}$$

其中, $D = \begin{bmatrix} 1 & \dots & \hat{p}_{k-1} \\ \vdots & \ddots & \vdots \\ \hat{p}_{k-1} & \dots & 1 \end{bmatrix}$, $D_k = \begin{bmatrix} 1 & \dots & \hat{p}_1 \\ \vdots & \ddots & \vdots \\ \hat{p}_{k-1} & \dots & \hat{p}_k \end{bmatrix}$, k 为滞后系数。

其次是数据差分处理。由于模型只能处理平稳数据, 故在面对非平稳数据时采用差分处理。运用差分处理时能用低阶满足就不需要用高阶处理, 原因是差分的过度使用会导致原始数据信息的丢失, 以下是差分得出的数学表达式:

$$Z_t = y_t - y_{t-1}, t = 1, 2, 3 \dots \quad (6.1.4)$$

其中, Z_t 表示差分后的数据; y_t 表示表示原始的物流路线负载值。

接着运用模型定阶。根据前面的分析可知, ARIMA(p, d, q) 模型中包含了自回归系数、差分次数以及移动平均项阶数。在运用 ARIMA 模型之前要先确定这三个变量的参数, 且根据所做差分处理的阶数确定该差分阶数; 根据自回归系数和移动平均项阶数并结合 AIC 和 BIC 标准, 选择出最小的参数。

以下是运用 AIC 准则选择最小参数的数学表达式:

$$AIC = -2\ln(L) + 2N \quad (6.1.5)$$

其中, L 为模型的似然函数; N 为模型中未知参数的个数。

紧接着是有效性检验。有效性检验主要包括对模型的显著性检验和对参数的显著性检验。对模型的显著性检验是根据检验残差序列判断其是否为白噪声序列, 利用 LB 检验统计量作为残差序列有效检验标准。

根据 ARIMA 模型, 首先做出检验假设。

$$H_0: \rho_1 = \rho_2 = \dots = \rho_m = 0, \forall m \geq 1$$

$$H_1: \text{至少存在某个 } \rho_k \neq 0, \forall m \geq 1, k < m \quad (6.1.6)$$

在此假设基础上构建 LB 统计量如下:

$$LB = n(n+2) \sum_{k=1}^m \frac{\hat{\rho}_k^2}{n-k} \sim X^2(m), \forall m > 0 \quad (6.1.7)$$

其中, $\hat{\rho}_k^2$ 是自相关系数; n 是项数; m 是总项数。当第一个假设成立时, 说明模型是显著有效的; 当第二个假设成立时, 说明残差序列中还在线性关系, 模型不显著有效。针对参数显著性检验主要是为了检验模型中所有参数是否是显著非零的, 通过删除不显著的参数来精简模型。针对模型参数做出如下假设, 其中 β_j 是参数。

$$H_0: \beta_j = 0 \quad (6.1.8)$$

$$H_1: \beta_j \neq 0 \quad (6.1.9)$$

参数显著性检验的统计量为

$$T = \sqrt{n-m} \frac{\hat{\beta}_j - \beta_j}{\sqrt{Q(\beta)}} \sim t(n-m) \quad (6.1.10)$$

其中, $Q(\beta)$ 是残差和方程, 当第一个假设成立时, 需要删除这个参数; 当第二假设成立时, 说明参数是显著的, 不需要删除。

最后进行 ARIMA 模型预测。通过上述步骤确定模型，使用确定好的模型对物流线路负载时间序列数据进行预测，得到物流路线负载预测结果为 $(\overline{A_t})$ 。

$$\overline{A_t} = (\sum_{i=1}^p \varphi_i S_{t-i} - \sum_{j=1}^q Z_{t-j}) \quad (6.1.11)$$

其中， S_t 表示差分平稳序列； Z_t 表示随机误差； p 表示自回归系数； q 表示移动平均数。

使用 python 自动优化 ARIMA 参数对 1049 条路线未来货量进行预测。以下是 14→10, 20→35, 25→62 三条路线的结果展示。

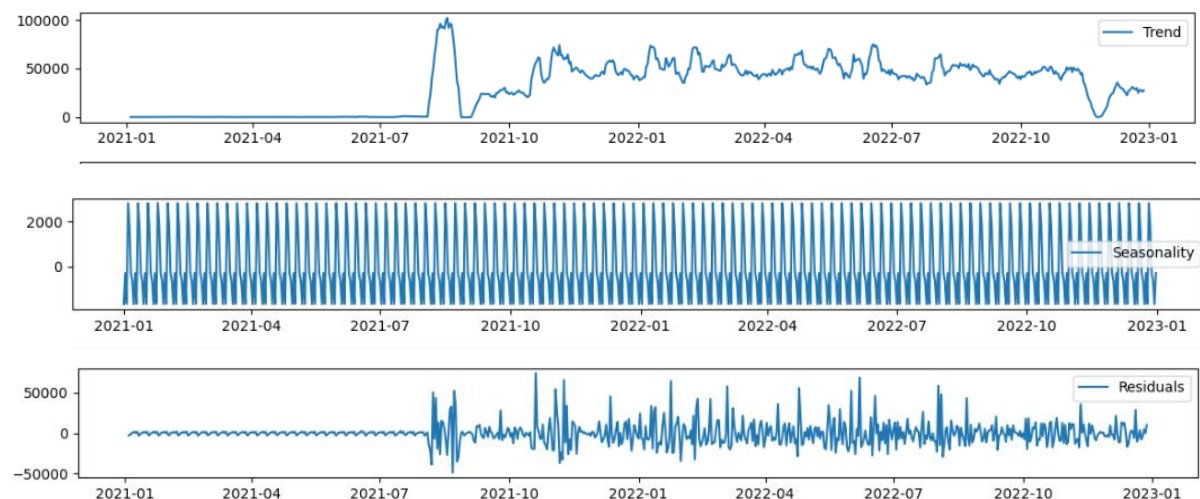


图 6.1.2 路线 14→10 的 ARIMA 模型季节性效应、趋势、随机波动拆分图

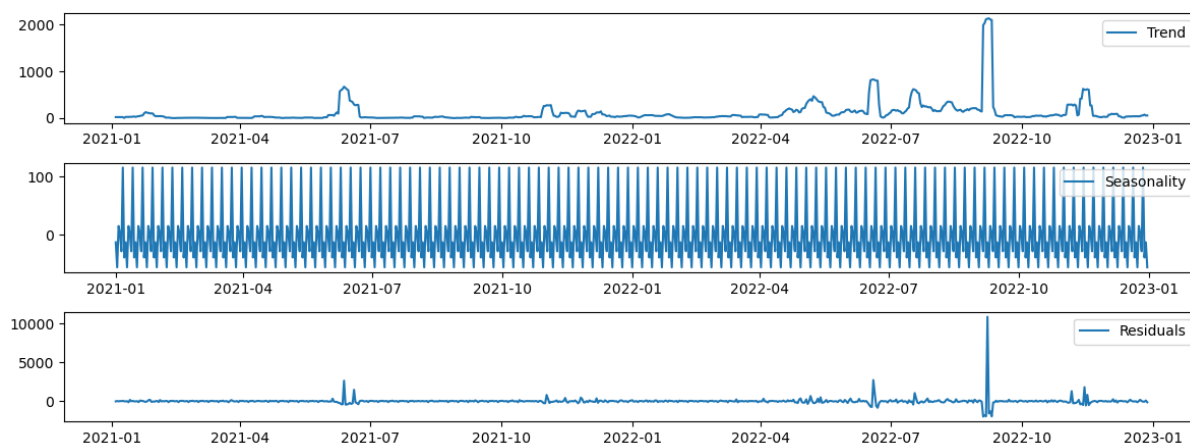


图 6.1.3 路线 20→35 的 ARIMA 模型季节性效应、趋势、随机波动拆分图

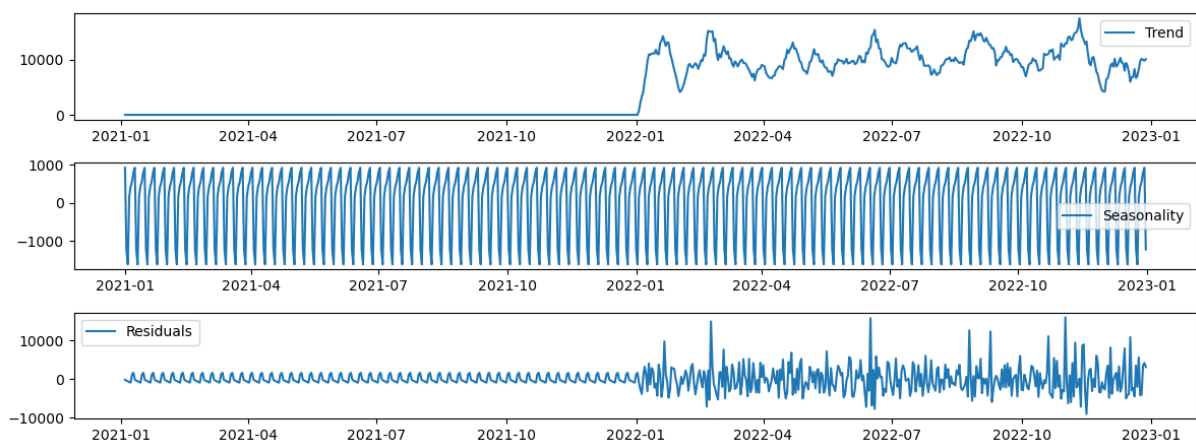


图 6.1.4 路线 25→62 的 ARIMA 模型季节性效应、趋势、随机波动拆分图
绘制模型残差自相关图 (ACF) 和残差偏自相关图 (PACF) 如下。

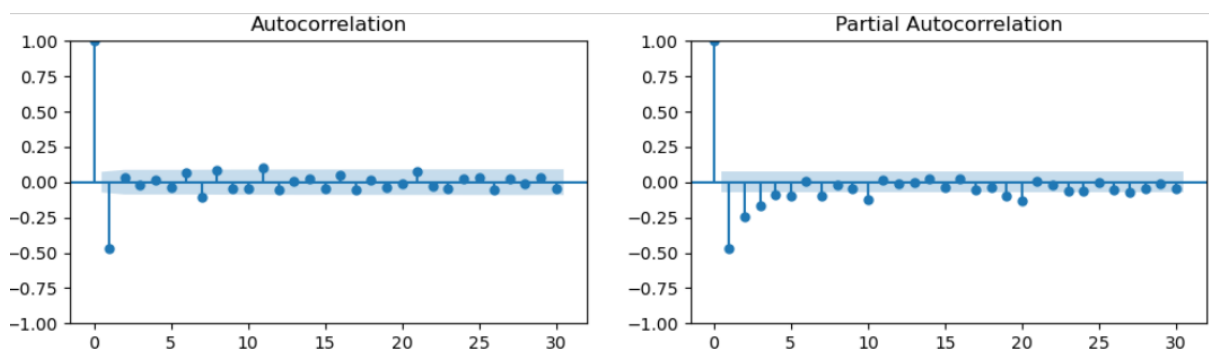


图 6.1.5 路线 14→10 残差自相关图 (ACF) 和残差偏自相关图 (PACF)

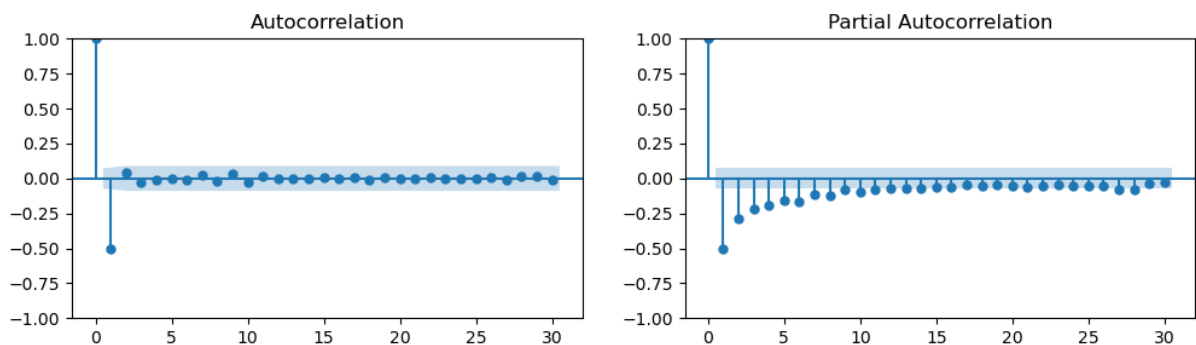


图 6.1.6 路线 20→35 残差自相关图 (ACF) 和残差偏自相关图 (PACF)

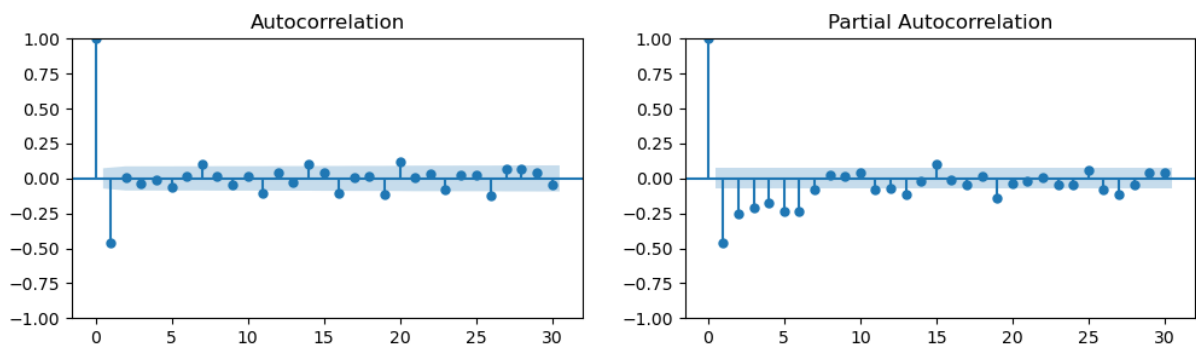


图 6.1.7 路线 25→62 残差自相关图 (ACF) 和残差偏自相关图 (PACF)
时间序列模型的原始数据图、模型拟合值、模型预测值如下。

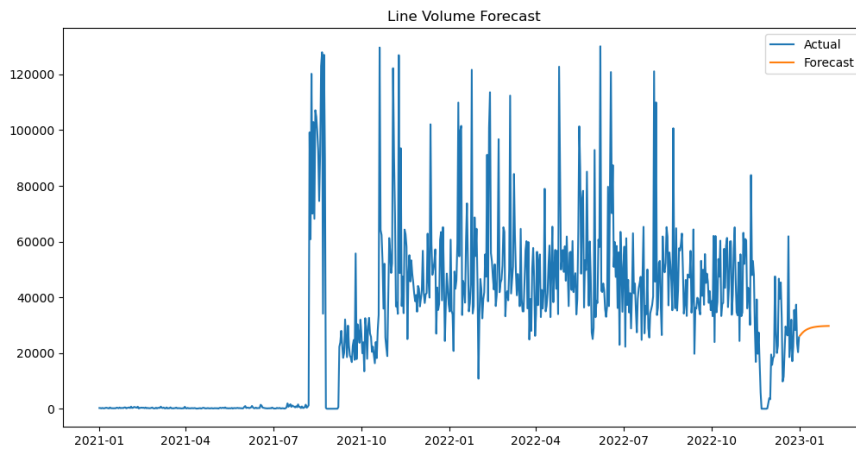


图 6.1.8 路线 14→10 时间序列图

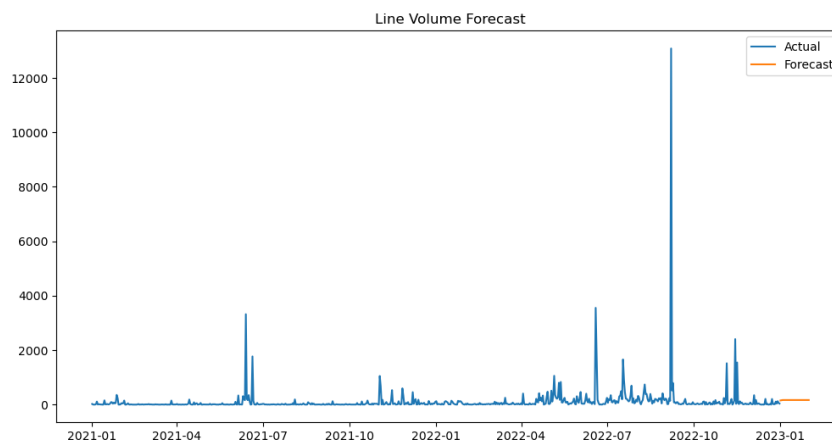


图 6.1.9 路线 20→35 时间序列图

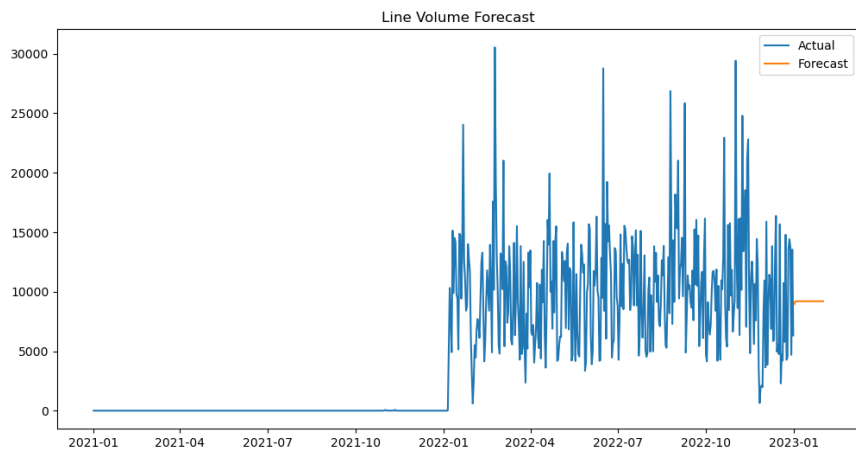


图 6.1.10 路线 25→62 时间序列图

下面是路线 14→10、路线 20→35、路线 25→62 的 ARIMA 模型检验表。

表 6.1.1 路线 14→10 的 ARIMA 模型 (0, 1, 1) 检验表

项	符号	值
	Df Residuals	727
样本数量	N	730
Q 统计量	Q6(P 值)	0.009(0.925)
	Q12(P 值)	1.671(0.947)
	Q18(P 值)	16.876(0.154)
	Q24(P 值)	32.274(0.020**)
	Q30(P 值)	42.282(0.012**)
信息准则	AIC	16333.069
	BIC	16346.844
拟合优度	R ²	0.623

注：***、**、*分别代表 1%、5%、10%的显著性水平

表 6.1.2 路线 25→62 的 ARIMA 模型 (1, 1, 1) 检验表

项	符号	值
	Df Residuals	726
样本数量	N	730
Q 统计量	Q6(P 值)	0.009(0.922)
	Q12(P 值)	8.402(0.210)
	Q18(P 值)	47.352(0.000***)
	Q24(P 值)	82.064(0.000***)
	Q30(P 值)	107.901(0.000***)
信息准则	AIC	13946.922
	BIC	13965.289
拟合优度	R ²	0.682

注：***、**、*分别代表 1%、5%、10%的显著性水平

基于 AIC 信息准则自动寻找最优参数，路线 14→10、路线 20→35、路线 25→62 的模型结果分别为 ARIMA 模型 (0, 1, 1)、ARIMA 模型 (2, 1, 1)、ARIMA 模型 (1, 1, 1)，根据检验表可得，从 Q 统计量结果分析可以得到，DC14→DC10 的 Q6 在水平上不呈现显著性，不能拒绝模型的残差为白噪声序列的假设，同时模型的拟合优度 R² 为 0.623，模型表现较为良好，模型基本满足要求；DC25→DC62 的 Q6 在水平上不呈现显著性，不能拒绝模型的残差为白噪声序列的假设，同时模型的拟合优度 R² 为 0.682，模型表现较为良好，模型基本满足要求。

由于货量单位为 1，因此将预测结果四舍五入得到模型预测结果如下。

表 6.1.3 路线 14→10 的部分 ARIMA 时间序列预测结果

预测值	
阶数（时间） 预测结果	
1	25898
2	25933
3	25968

4	26003
5	26038
6	26072
7	26107
8	26142
9	26177
10	26212

表 6.1.4 路线 20→35 的部分 ARIMA 时间序列预测表

预测值	
阶数（时间）	预测结果
11	232
12	232
13	232
14	233
15	233
16	233
17	234
18	234
19	234
20	235

表 6.1.5 路线 25→62 的部分 ARIMA 时间序列预测表

预测值	
阶数（时间）	预测结果
21	9539
22	9551
23	9564
24	9577
25	9590
26	9603
27	9615
28	9628
29	9641
30	9654
31	9667

6.1.2 时间序列 LSTM 模型的构建与求解

LSTM 预测模型构建过程如图所示。从图中可知 LSTM 预测模型构建分为数据预处理、模型参数设置、模型训练和结果输出。

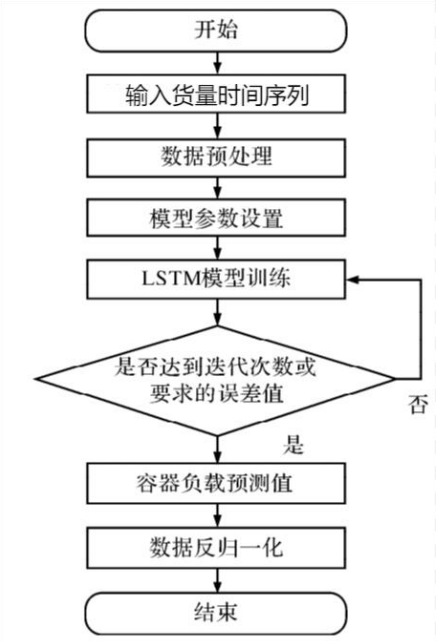


图 6.1.11 LSTM 模型的构建

首先是数据预处理。原始的物流路线负载时间序列数据不适合直接被预测模型使用。因此，在使用数据之前需要对数据进行预处理，常用的预处理方法是将时间序列数值范围限定在 $[0, 1]$ 之间，即数据归一化处理。

然后对模型参数进行设置。LSTM 模型中有训练窗口数大小、隐藏层节点数以及训练迭代次数参数，并通过调整参数值训练模型。模型激活函数的选取至关重要，考虑到 ReLu 模型具有明显的优势，本文选用 ReLu 激活函数。

接着模型训练和模型结果输出。将物流路线负载数据输入到模型中进行训练，当达到迭代次数时，训练结束。使用训练好的模型实现物流路线负载数据的预测，将预测出来的结果进行反归一化，得到最终的预测值 \hat{L}_t 。

$$\hat{L}_t = f_l(y_{t-1}, y_{t-2}, \dots, y_{t-n}) \tag{6.1.12}$$

式中， y 为的是物流路线负载原始时间序列； f_l 为 LSTM 模型代表的函数关系； n 为窗口大小。

由于货量单位为 1，因此将预测结果四舍五入得到模型预测结果。

6.1.3 时间序列 ARIMA 与 LSTM 混合模型的构建与求解

本文构建的货量预测模型为如图所示。

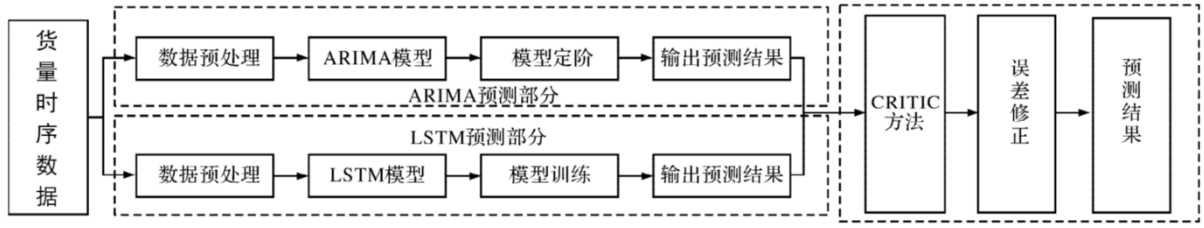


图 6.1.12 ARIMA-LSTM 预测模型

由图可知，将货量使用时间序列输入到 ARIMA 模型中，使得线性货量预测值；比较原始数据和线性拟合数据，得到拟合误差序列，再利用 LSTM 模型预测拟合误差序列；再将线性拟合预测值与非线性拟合误差预测值相加，得出混合模型的最终预测结果。

利用 CRITIC 客观赋权法对这两种预测结果相互融合，并且使用误差修正提升预测精度。CRITIC 是一种客观的权重赋权方法，通过各指标概率和各指标权重计算出各模型的权重，能更好的挖掘各单一模型中的有用信息^[1]。其中这里的指标指的是评价预测结果的误差评价指标，本文中采取了 4 种误差评价指标。

首先，根据对指标权重的计算，得到各指标所包含的信息量 C_j 的有关表达式如下：

$$C_j = \sigma_j \sum_{i=1}^m (1 - t_{ij}), j = 1, 2, \dots, k \quad (6.1.13)$$

其中， σ_j 表示的是第 j 项指标的标准差，又本文采用了 4 种误差评价指标，故 $k=4$ ； t_{ij} 为评价指标 i 与 j 间的相关系数， m 则表示为模型的个数且 m 的取值为 2。信息量 C_j 取值越大，则说明指标所包含的信息量也就越大。

根据各指标所包含的信息量 C_j 计算得出各指标的权重 W_j 为

$$W_j = \frac{C_j}{\sum_{j=1}^k C_j}, j = 1, 2, 3, \dots, k \quad (6.1.14)$$

接着根据指标权重 W_j 和指标率 p_{ij} 求出各模型的权重 P_i 。

$$AIC = -2\ln(L) + 2N \quad (6.1.15)$$

$$p_{ij} = \frac{r_{ij}}{\sum_{i=1}^m r_{ij}} \quad (6.1.16)$$

其中， r_{ij} 表示的是标准差评价指标值。

最后，利用各模型权重融合各模型的预测结果如下。

$$value_t = P_{ARIMA} * \hat{A}_t + P_{LSTM} * \hat{L}_t \quad (6.1.17)$$

其中， P_{ARIMA} 为求出的 ARIMA 模型的权重； P_{LSTM} 为求出的 LSTM 模型的权重； \hat{A}_t 为 ARIMA 模型的预测值； \hat{L}_t 为 LSTM 模型的预测值。

计算得出的权重值分别为 $P_{ARIMA} = 0.5$ ， $P_{LSTM} = 0.5$ 。得到组合预测结果后，通过组合预测结果和真实值求出误差并且对误差进行修正，进一步提升预测结果精度。

得到最终预测结果如下。

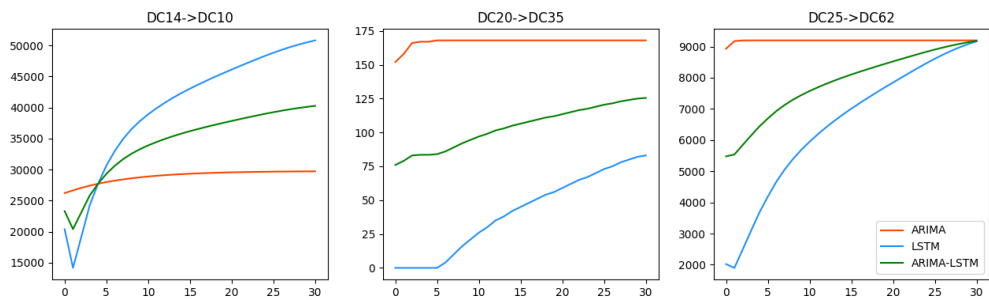


图 6.1.13 ARIMA、LSTM 和 ARIMA-LSTM 组合模型结果对比图

可以看到，ARIMA-LSTM 组合模型可以较好地均衡解释了 ARIMA 和 LSTM 模型的预测结果。

表 6. 1. 6 时间序列预测表

ARIMA-LSTM 时间序列预测值			
时间	路线 14→10 预测结果	路线 20→35 预测结果	路线 25→62 预测结果
1	23306.5	76	5476.5
2	20413.5	79	5536.5
3	23133.5	83	5844.5
4	25831.5	83.5	6145.5
5	27722	83.5	6439.5
6	29325	84	6691.5
7	30612.5	86	6925
8	31685	89	7123
9	32558.5	92	7293
10	33280	94.5	7440.5
11	33898.5	97	7572
12	34447	99	7693
13	34942	101.5	7805.5
14	35393.5	103	7910.5
15	35809.5	105	8009.5
16	36192	106.5	8102.5
17	36549.5	108	8191
18	36891.5	109.5	8277
19	37220	111	8361
20	37534.5	112	8443
21	37835.5	113.5	8522.5
22	38135.5	115	8601.5
23	38429	116.5	8680
24	38714	117.5	8757
25	38988	119	8832
26	39247.5	120.5	8903.5
27	39492	121.5	8970.5
28	39717.5	123	9032.5
29	39920.5	124	9088.5
30	40103.5	125	9138.5
31	40270.5	125.5	9182.5

6.2 问题二模型的建立与求解

6.2.1 物流路线配送能力函数的建立

首先，我们计算出每一条线路的平均载荷量，将每一条线路在不同日期下的载荷量中的最大值作为此线路运输能力的上限即负荷。

路线平均载荷与该路线的最大负荷量的比值反映了该路线对于货物激增情况下的弹性，比值越小代表对于货物激增现象的反应能力越强，物流的承压能力越强，呈负相关关系。因此定义物流承压能力 P_{ij} 函数方程如下，其中 P_{ij} 越大，物流路线承压能力越强。

$$P_{ij} = \frac{M_{ij}}{A_{ij}} (i = 1, 2, \dots, 81; j = 1, 2, \dots, 81; i \neq j) \quad (6.2.1)$$

定义物流路线稳定能力 S_{ij} 函数方程如下，其中 S_{ij} 越大代表该路线最大载货量占该物流点最大出货量的比例越高，该物流点在选择路线时偏向性越大，物流路线越稳定。

$$S_{ij} = \frac{M_{ij}}{O_{ij}} (i = 1, 2, \dots, 81; j = 1, 2, \dots, 81; i \neq j) \quad (6.2.2)$$

问题二要求在物流网络规划时考虑在满足包裹正常运转的前提下物流路线的负荷尽可能均衡，变化的物流路线尽可能少，因此，物流的承压能力与物流路线稳定能力是判断其路线规划的重要指标。通过定义物流线路配送能力系数函数，将系数计算结果值作为系数定义功能坐标，可以将物流路线的承压能力与稳定能力直接转化为路径长度，使得最优路径满足物流路线的负荷尽可能均衡，变化的物流路线尽可能少的约束。路径越短代表物流配送能力越强。因此物流线路配送能力系数函数定义如下，其中， $(i=1, 2, \dots, 81; j=1, 2, \dots, 81; i \neq j)$ 。

$$D_{ij} = \frac{1}{P_{ij} \times S_{ij}} = \frac{O_{ij}}{M_{ij}} \times \frac{A_{ij}}{M_{ij}} = \frac{O_{ij} \times A_{ij}}{M_{ij}^2} \quad (6.2.3)$$

6.2.2 MDS 算法的建立与功能坐标的求解

假设已有的 2 年货量数据中的极端数据是最优值，可以作为路径长度的参考。根据物流路线配送能力系数使用 MDS 算法构建功能坐标，其中，令场所 A 到场所 B 的功能坐标距离为场所 A 到场所 B 与场所 B 到场所 A 的物流路线配送能力系数平均值。由于系数之间差值较大，对数据首先做标准化处理再计算坐标，对于两点之间不存在的路线，我们看作它的物流配送能力较弱，因此将其定义为已有系数的最大值，即为 1。

MDS 算法的基本思想是将高维坐标中的点投影到低维空间中，保持点彼此之间的相似性尽可能不变^[2]。

首先计算原始空间中数据点的距离矩阵。设降维后第 i 个点与第 j 个点的距离如下公式。

$$d_{ij} = \|z_i - z_j\|^2 = \|z_i\|^2 + \|z_j\|^2 - 2z_i^T z_j \quad (6.2.4)$$

假设 z 维空间中的实例点是中心化（所有点在任意一维上的坐标和为 0）的，即：

$$\sum_{i=1}^N z_i = 0 \quad (6.2.5)$$

公式(4)左右两边在 i 上求和得：

$$\sum_{i=1}^N d_{ij} = \sum_{i=1}^N \|z_i\|^2 + N \|z_j\|^2 - 2 \sum_{i=1}^N z_i^T z_j = \sum_{i=1}^N \|z_i\|^2 + N \|z_j\|^2 \quad (6.2.6)$$

同理，左右两边在 j 上求和得：

$$\sum_{j=1}^N d_{ij} = \sum_{j=1}^N \|z_j\|^2 + N \|z_i\|^2 - 2 \sum_{j=1}^N z_j^T z_i = \sum_{j=1}^N \|z_j\|^2 + N \|z_i\|^2 \quad (6.2.7)$$

公式（6）左右两边再在 j 上求和得：

$$\sum_{i=1}^N \sum_{j=1}^N d_{ij} = \sum_{i=1}^N \sum_{j=1}^N \|z_i\|^2 + N \sum_{j=1}^N \|z_j\|^2 = 2N \sum_{i=1}^N \|z_i\|^2 \quad (6.2.8)$$

计算内积矩阵 B ：

$$B = Z^T Z \quad (6.2.9)$$

则

$$b_{ij} = z_i^T z_j = -\frac{1}{2}(d_{ij} - \|z_i\|^2 - \|z_j\|^2) \quad (6.2.10)$$

由公式(6)(7)可得：

$$\|z_i\|^2 = \frac{1}{N} \sum_{j=1}^N d_{ij} - \frac{1}{N} \sum_{j=1}^N \|z_j\|^2 \quad (6.2.11)$$

$$\|z_j\|^2 = \frac{1}{N} \sum_{i=1}^N d_{ij} - \frac{1}{N} \sum_{i=1}^N \|z_i\|^2 \quad (6.2.12)$$

$$\begin{aligned} z_i^T z_j &= -\frac{1}{2} \left(d_{ij} - \frac{1}{N} \sum_{j=1}^N d_{ij} + \frac{1}{N} \sum_{j=1}^N \|z_j\|^2 - \frac{1}{N} \sum_{i=1}^N d_{ij} + \frac{1}{N} \sum_{i=1}^N \|z_i\|^2 \right) \\ &= -\frac{1}{2} \left(d_{ij} - \frac{1}{N} \sum_{j=1}^N d_{ij} - \frac{1}{N} \sum_{i=1}^N d_{ij} + \frac{2}{N} \sum_{i=1}^N \|z_i\|^2 \right) \end{aligned} \quad (6.2.13)$$

$$\frac{2}{N} \sum_{i=1}^N \|z_i\|^2 = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N d_{ij} \quad (6.2.14)$$

$$b_{ij} = z_i^T z_j = -\frac{1}{2} \left(d_{ij} - \frac{1}{N} \sum_{j=1}^N d_{ij} - \frac{1}{N} \sum_{i=1}^N d_{ij} + \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N d_{ij} \right) \quad (6.2.15)$$

对矩阵 B 进行特征值分解，获得特征值矩阵 A 和特征向量矩阵 V 。由于 B 是一个是对称矩阵，因此对矩阵 B 进行特征分解可以得到：

$$B = V A V^T \quad (6.2.16)$$

其中， A 是特征值矩阵， V 是特征向量矩阵，所以取特征值矩阵最大的前 Z 项及其对应的特征向量。

$$Z = V_Z A_Z^{\frac{1}{2}} \quad (6.2.17)$$

坐标构建结果如图所示，坐标距离取值位于 0-1 之间，点与点之间坐标越近代表两点间物流路线配送能力越强。

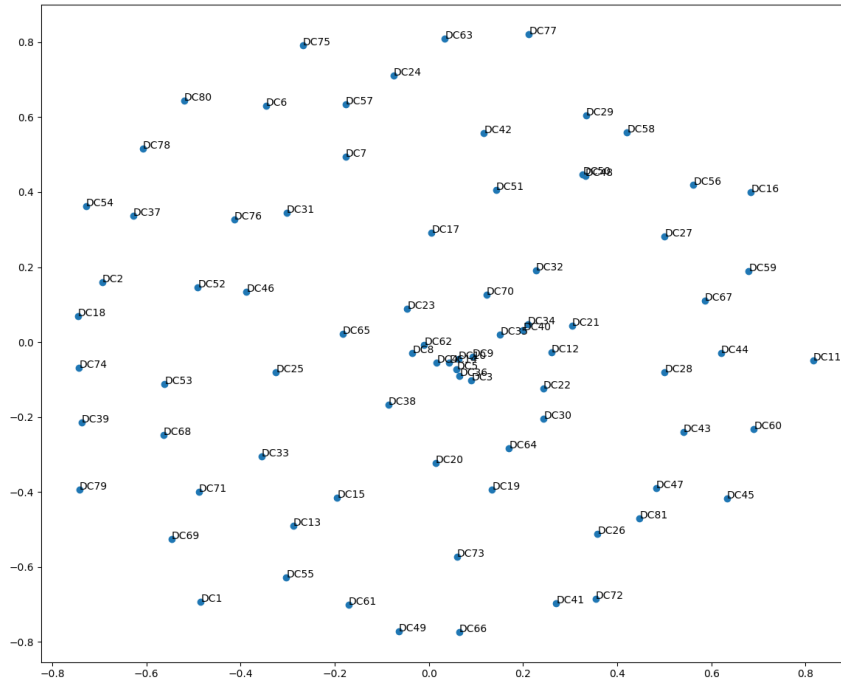


图 6.2.1 各物流点功能坐标位置图

图 6.2.2 所有物流点及路线网络图

6.2.3 优化函数的构建

物流场地 DC5 于 2023-01-01 开始关停，将 DC5 相关路线的货物分配到其他路线，设 DC5 未关停前运输到 DC5 的物流点为 DC_i (i=1,2,⋯,n), DC5 运输的物流点为 DC_j (j=1,2,⋯,n), x_{ij}^0 为 D5 关停前 DC_i 到 DC_j 的货运量, x_{ij}^1 为 D5 关停后 DC_i 到 DC_j 的货运量。

目标函数的构建

①C5 的相关路线的货量分配到其他路线后，使 DC5 关停前后 DC_i→DC_j 的货量变化最小。即

$$\text{MIN } \sum_i^n \sum_j^n |x_{ij}^1 - x_{ij}^0| \quad (6.2.18)$$

②所有线路上的货物数量尽可能均衡

$$\text{MIN}(\max_{i=1,2,\dots,n} \left(x_{ij}^1 - \frac{\sum_i^n \sum_j^n x_{ij}^1}{n} \right) - \min_{i=1,2,\dots,n} \left(x_{ij}^1 - \frac{\sum_i^n \sum_j^n x_{ij}^1}{n} \right)) \quad (6.2.19)$$

③使 DC5 关停前后货量发生变化的线路尽可能少，记变化的路线有 n 条，即

$$\text{MIN } c_i \quad (6.2.20)$$

约束函数的构建

为保持各条线路的工作负荷尽可能均衡，需要将每条线路的总货物数量限制在一个相对稳定的范围内，即所有线路的工作负荷尽量相近。因此，需要添加以下约束条件：

①所有线路上的货物数量必须满足不能超过路线的最大负荷，即

$$0 \leq \sum_i^n \sum_j^n x_{ij}^1 \leq M_{ij} \quad (6.2.21)$$

②设关停 DC5，线路上的货量分配到其他线路的比例为 p_{ij} ，则其他线路上的实际包裹量为 $p_{ij}x_{DC5} + x_{ij}$

$$p_{ij}x_{DC5} + x_{ij} \leq M_{ij} \quad (6.2.22)$$

③DC5 相关线路上的货物数量分配到其他线路上的总量等于 DC5 相关线路上的货物

$$\sum_{j=1}^n \sum_{i=1}^n x_{ij}^0 = \sum_{j=1}^n \sum_{i=1}^n x_{ij}^1 \quad (6.2.23)$$

6.2.4 多目标蚁群算法的选择

如图是一个简单的具体的物流配送线路实例，假设有 1 个物流中心、6 个配送点和 3 辆车。左边的 3 个客户点的配送任务走配送线路 1，下边的 1 个客户点和右边的 2 个客户点分别走线路 2 和线路 3；3 辆配送车辆分别从配送中心出发进行配送物品，按顺序遍历指定线路上的客户点，最后返回到配送中心^[3]。

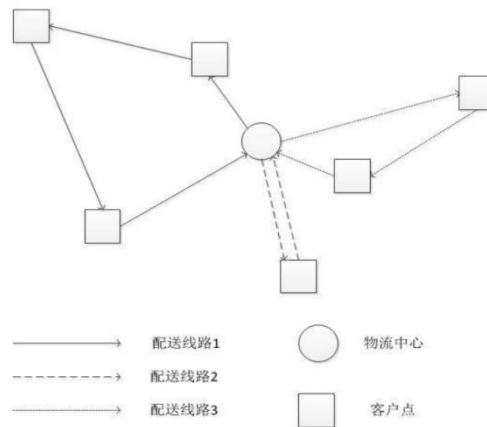


图 6.2.3 物流配送示例图

蚁群算法是受自然界蚂蚁觅食行为的启发而产生的一种自然模拟进化算法。蚂蚁从蚁穴出发寻找食物源的大概过程是：刚开始的时候，蚂蚁选择路径是随机的，但是后来路径的选择会随着寻找食物过程的继续而适应性地搜索新的路径。其中的主要原因是蚁群中的蚂蚁在寻找食物或者寻找回巢的路途中，会在它们经过的地方释放出“信息素”这一化学物质以保持间接的异步的联系，而路径上的信息素会随着时间的流逝，按照一定的比例逐渐减少，同时信息素的浓度还与路径长度有关，路径越短则信息素浓度越大；在一定的时间内，若某一路径上经过的蚂蚁越多，则其积累的信息素浓度越强，那么在下一个时间内被其他的蚂蚁选中的可能性也就越大，如此循环下去形成正反馈效应，直到所有的蚂蚁都走最短的那一条路径为止^[4]。此外，并非后面的蚂蚁一定会选择信息素浓度高的路径，只是选择浓度高的路径的概率比较大而已，这就为蚂蚁扩大搜索范围提供了可能性。蚂蚁觅食行为图具体如图所示。

如下图为蚁群觅食行为图，图一到图四分别为所有蚂蚁在 nest 处周围没有信息素的示意图，开始觅食时蚂蚁两条路径概率相同时的示意图，蚂蚁选择较短路径提前到达示意图以及由于较短路径信息素加强使得所有蚂蚁选择较短路径概率大于较长路径的示意图。

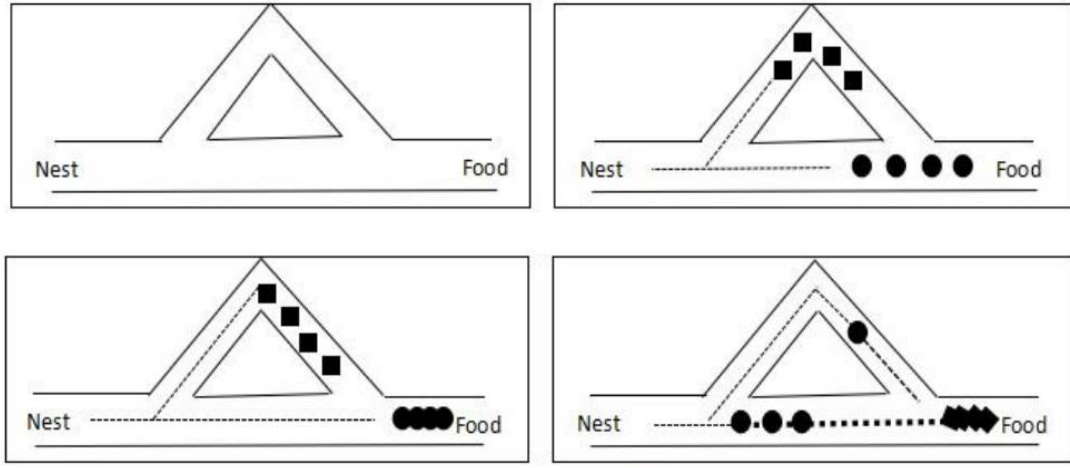


图 6.2.4 蚁群觅食行为图

要实现人工模仿蚂蚁觅食过程，首先需要先将其符号化表示：已知蚁群中蚂蚁的数量是 m ， $d_{ij}(i, j=1, 2, \dots, n)$ 表示物流点 i 和物流点 j 之间的距离， $\tau_{ij}^{(t)}$ 表示 t 时刻 i 和 j 之间的剩余信息素， η_{ij} 表示启发性因子，由某种启发式算法确定，表示蚂蚁从 i 移动到 j 的期望度，在本文中表示路径的能见度，即 $1/d_{ij}$ ， $allowed_k$ 表示蚂蚁 k 下一步允许选择的物流点。

令各个物流点之间在开始时刻的信息素为某一具体常数，设 $\tau_{ij}^{(0)} = C$ ，然后计算各路径上的信息素浓度大小，筛选出计算结果最大的路径作为蚂蚁 $k(k=1, 2, \dots, m)$ 下一个要前进的目标， $p_{ij}^k(t)$ 表示在 t 时刻蚂蚁 k 由 i 转移到 j 的概率，具体公式如下。

$$P_{ij}^k(t) = \frac{\tau_{ij}(t)^{\alpha} \eta_{ij}^{\beta}}{\sum_{k \in allowed} \tau_{ij}(t)^{\alpha} \eta_{ij}^{\beta}} (j \in allowed)$$

(6.2.24)

式中， α 是信息启发因子，表示蚂蚁在运动过程中所积累的信息， β 为期望启发因子，表示 η_{ij} 在蚂蚁面临路径选择时所起的不同影响。提高蚁群算法效率的有效措施之一就是具体问题具体分析，对应到此式中就是确定恰当的 α 和 β 的值。

人工蚁群系统中用 $tabu_k(k=1, 2, \dots, m)$ 来标记蚂蚁 k 已经走过的点。最初集合 $tabu_k$ 中仅包含蚂蚁刚开始所处位置，程序的持续运行使得 $tabu_k$ 中的点数逐渐增多。参数 $1-\rho$ 表示信息消逝程度，即时间的流逝会使得蚂蚁走过的路上积累的信息素按一定的速度消逝。为避免残留的信息素过多导致残留信息淹没启发信息，因此，需要在每只蚂蚁完成对所有物流点的一次遍历，也就是一个循环结束后，需要对路径上信息素的浓度根据下列公式进行更新。

$$\tau_{ij}(t+n) = (1+\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (6.2.25)$$

$$\Delta\tau(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (6.2.26)$$

上列公式中， ρ 表示信息素的挥发系数， $\rho \in (0, 1)$ 。 $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在本次循环中留在路径 ij 上的信息量， $\Delta\tau_{ij}$ 表示 m 只蚂蚁在本次循环中留在路径 ij 上的信息量。 $\Delta\tau_{ij}^k$ 的具体计算表达式，如式(21)。其中 L_k 表示在本次循环中第 k 只蚂蚁所走过的路径的总长度， Q 是表示信息素强度的一个常数，影响算法搜索出最优解所用时间。

$$\Delta\tau_{ij}^k = \frac{Q}{L_k} (\text{第 } k \text{ 只蚂蚁选择的路径}(i,j), t = t + n) \quad (6.2.27)$$

根据上述蚁群算法模型的基本描述可以得到以下基本蚁群算法的流程图。

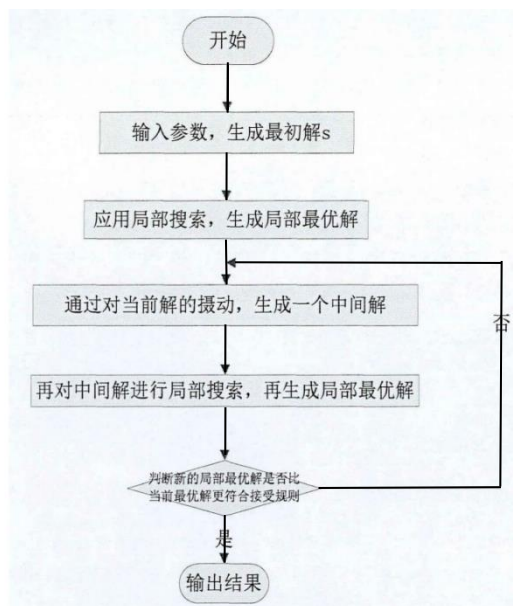


图 6.2.5 迭代局部搜索流程

由于物流网络路线错综复杂, 因此本文假设 DC5 关闭前后只影响 DC5 上一级物流点到 DC5 和 DC5 到 DC5 下一级物流点的运输路线, 不考虑 DC5 关闭对其他无关物流路线的影响。因此 DC5 货物运输的上一个物流点即为起点, DC5 货物运输线路的下一个物流点即为货物终点。

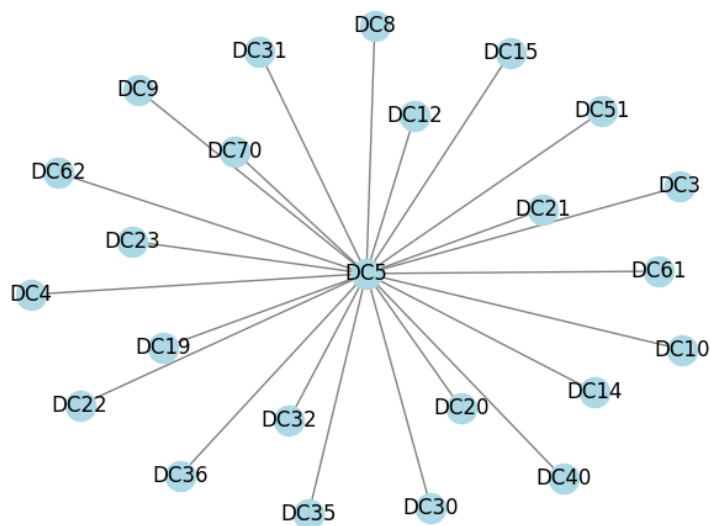


图 6.2.6 以 DC5 为发货点的相关路线图

在本次问题中, 我们使用多目标蚁群算法来解决货运量调配问题^[5]。蚁群算法的过程包括初始化信息素矩阵、计算启发式信息矩阵、创建蚂蚁、让蚂蚁选择路径、更新信息素矩阵和挥发信息素等步骤。其中, 信息素矩阵用于表示路径的优劣, 启发式信息矩阵用于表示路径的可行性和优劣, 蚂蚁通过选择概率来选择下一个点, 选择的过程中考

虑了信息素和启发式信息的影响，选择完路径后更新信息素矩阵，以便引导下一次的蚂蚁选择。本问需要根据具体问题进行参数的调整，包括信息素重要程度因子、启发式因子、信息素挥发因子、常量因子、蚂蚁数量、迭代次数等。

具体来说，我们需要根据给定的物流点坐标、原货运量和不可用站点，找到与 DC5 相关的路径，使得，蚂蚁在满足以下约束下在撤销 DC5 后找到最优的路径。

①读取两个 excel 表格，表 1 有场所、x、y、处理能力 4 个标签，表 2 有场所 1、场所 2、货量、路线载荷四个标签。场所间的来往关系是场所 1 到场所 2，如 DC1→DC8，场所的坐标是 (x, y)。

②初始化蚂蚁的位置，蚂蚁的初始位置是场所 1，终止位置是场所 2。

③计算撤销物流场地 DC5 后的目标函数：

目标 1：DC5 相关线路的货量分配到其他线路使所有包裹尽可能正常流转。

目标 2：所有线路上的货物数量尽可能均衡。

目标 3：DC5 关停前后货量发生变化的线路尽可能少。

④约束函数：

约束 1：路线货物数量不能超过路线的最大负荷。

约束 2：撤销 DC5 后其被分配到货物的线路的数量不能超过线路的最大负荷。

约束 3：DC5 相关线路上的货物数量分配到其他线路上的总量等于 DC5 相关线路上的货物。

⑤初始化蚂蚁的信息素和可行解集合。

⑥迭代搜索过程：

蚂蚁根据信息素和启发式函数选择下一个场所，更新蚂蚁的路径和载荷；

检查路径是否符合约束条件，如果不符合，则重新选择路径；

更新信息素和可行解集合；

检查是否达到停止条件，如果没有达到，则返回步骤 1。

⑦最终输出可行解集合，根据目标函数进行筛选，选择最优解。

对信息素矩阵的更新规则，我们做如下定义：

如果当前的蚂蚁的路线有利于目标函数，则提高这条路线的概率。

$$\min Var(P) \quad (6.2.28)$$

$$\min \sum_{i=1}^n (l_t - l'_t) \quad (6.2.29)$$

$$\min \sum c_i \quad (6.2.30)$$

其中, l_t 为当天的实际负载, l_t' 为蚁群模拟负载, $P = (l_1, l_2, \dots, l_n)$ 是有变化的路线的负载向量, 其中 $l_i = \frac{l_t}{l_{\max}}$ 。为使得 DC5 关停前后货量发生变化的线路尽可能少, 我们再考虑负载变化较大的路线总数 $\sum c_i$ 尽量少。

随后我们指定约束条件:

$$0 \leq l_t^{(i)'} \leq l_{\max}^{(i)} \quad (6.2.31)$$

首先我们使用实际货运量, 考虑经济周期相似, 使用 2022 年 1 月的数据。对蚁群算法进行参数调优, 使得蚁群的模拟结果接近历史值, 即:

$$\min \sum_{i=1}^n (\hat{x}_i - x_i) \quad (6.2.32)$$

其中, \hat{x}_i 表示模拟结果, x_i 表示真实值。参数调优结果如下:

```
alpha = 1    # 信息素重要程度因子
beta = 5     # 启发式因子
rho = 0.1    # 信息素挥发因子
Q = 100     # 常量因子
num_ant = 50 # 蚂蚁数量
num_iter = 100 # 迭代次数
```

图 6.2.7 参数调优结果

在问题二中, 我们关停了 DC5, 且不能新增路线。为了避免出现不可行解, 我们可以增加惩罚因子, 对经过 DC5 的路径进行惩罚。即修改这些路径的信息素, 让概率接近 0。设关停后 DC5 线路上的货量平均分配到其他线路的比例为 p (分流比例), 则其他线路上的每天的实际包裹量为:

$$l_t^{(i)'} = p^{(i)} \times l_t^{(DC5)} + l_t^{(i)} \leq l_{\max}^{(i)} \quad (6.2.33)$$

$$\sum p^{(i)} = 1 \quad (6.2.34)$$

利用多目标蚁群算法运算结果如下所示。

表 6.2.1 关停 DC5 后的路线网络负荷情况表

场地 1	场地 2	货量(件)	工作负荷%
1	2	142	97.51
1	6	228	95.97
2	3	178	97.25
2	4	179	96.22
2	6	143	96.04
3	4	194	97.06

3	8	239	97.68
3	9	162	95.09
4	5	175	96.57
4	6	76	77.62
4	7	89	97.85
4	8	138	95.59
5	6	129	92.21
5	7	82	82.31
5	9	68	66.33
6	7	190	96.57
6	9	167	95.74
7	8	138	97.08
8	9	212	97.64

表 6.2.2 关停 DC5 后货量改变的线路表

场地 1	场地 2	工作负荷%
7	3	97.61
23	9	96.87
37	10	95.23
71	14	96.26

由表可知，正常流转的情况下，关停 DC5 后导致货量发生变化的线路数为 4 条，即 DC7→DC3,DC23→DC9,DC37→DC10,DC71→DC14。各条路线的货量和工作负荷情况如上表所示。

6.3 问题三模型的建立与求解

6.3.1 多目标动态函数的建立

针对问题 2，需要建立一个动态调整的货物量数学模型，该模型需要在每天进行一次决策，决定路线的开启和关闭，在关停 DC9 的同时，需将 DC9 相关路线的货物分配到其他路线，使得所有包裹尽可能正常流传，并使由于 DC9 的关停而导致货量发生变化的线路尽可能少，且保持各条线路的工作负荷尽可能均衡。

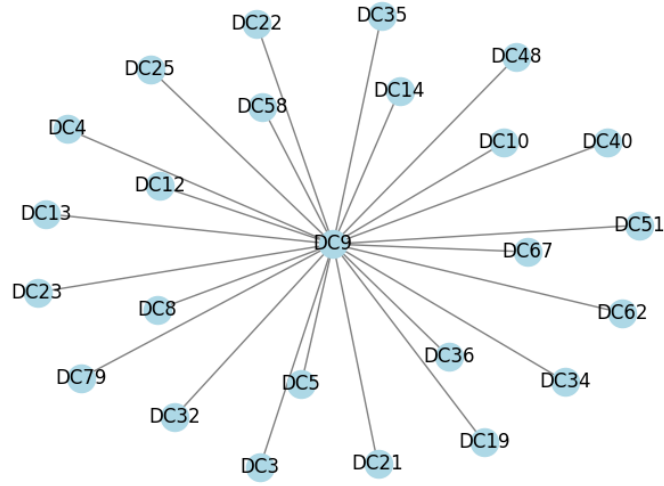


图 6.3.1 以 DC9 为发货点的相关路线图

定义路线是否开通符号说明：

$m_{ijt}=0$ 表示第 t 天从 i 到 j 未开通新的路线

$m_{ijt}=1$ 表示第 t 天从 i 到 j 开通新的路线

$n_{ijt}=0$ 表示第 t 天从 i 到 j 未关闭的路线

$n_{ijt}=1$ 表示第 t 天从 i 到 j 关闭的路线

首先是目标函数的构建。

①为使 DC9 关停前后货量发生变化的线路数尽可能少，应将所有的路线变量相加，使得总开通的路线数和总关闭的路线数量最小，即：

$$\text{MIN} \left(\sum_{i \in V} \sum_{j \in V} \sum_t^T m_{ijt} + \sum_{i \in V} \sum_{j \in V} \sum_t^T n_{ijt} \right) \quad (6.3.1)$$

②为了尽可能减少不能正常流转的包裹数量，定义一个变量 q_{ijt} 表示第 t 天从 i 到 j 滞留的货物量，本文的目标是使无法正常流转的货物量最少，即：

$$\text{MIN} \sum_{i \in V} \sum_{j \in V} \sum_t^T q_{ijt} \quad (6.3.2)$$

③为了保持各条线路的工作负荷尽可能均衡，定义变量 w_{ij} 表示线路 i 到 j 的工作负荷，即该线路的货量总和。为最小化所有线路的工作负荷不平衡度，可采用标准差的方法来度量负载均衡度，公式如下：

$$\text{MIN} \sqrt{\frac{\sum_{i \in V} \sum_{j \in V} (w_{ij} - \bar{w})^2}{n}} \quad (6.3.3)$$

④所有线路上的货物数量尽可能均衡。

$$\text{MIN} \left(\max_{i=1,2,\dots,n} \left(x_{ij}^1 - \frac{\sum_i^n \sum_j^n x_{ij}^1}{n} \right) - \min_{i=1,2,\dots,n} \left(x_{ij}^1 - \frac{\sum_i^n \sum_j^n x_{ij}^1}{n} \right) \right) \quad (6.3.4)$$

然后是约束函数的构建。

①新开线路的运输能力不能超过已有线路运输能力的最大值。

$$x_{ijt} \leq m_{ijt} * c_{ij} \quad (6.3.5)$$

其中 c_{ij} 为已有路线的最大值。 x_{ijt} 表示第 t 天从 i 到 j 开通路线运输能力。

②当关闭某个节点时，从该物流点出发的货物流量为0。

$$\sum_{i \in V} n_{ijt} * x_{ijt} = 0, \forall i, j \in V \quad (6.3.6)$$

③所有包裹均能正常流转，使得流入的总货物量减去流出的总货物量等于新增点流入的总货物量减去流出的新增点的总货物量。

$$\sum_{i \in V} x_{ijt} - \sum_{i \in V} x_{jit} = (1 - m_{ijt}) \sum_{i \in V} x_{ij(t-1)} + (1 - n_{ijt}) \sum_{i \in V} x_{ji(t-1)}, \forall i, j \in V \quad (6.3.7)$$

④每个时间段每个线路的货物数量为非负整数。

$$x_{ijt} \in Z \quad (6.3.8)$$

⑤所有线路上的货物数量必须满足不能超过路线的最大负荷，即：

$$0 \leq \sum_i^n \sum_j^n x_{ij}^1 \leq M_{ij} \quad (6.3.9)$$

同问题二，使用多目标蚁群算法求解，但与问题二不同的是，问题二的蚁群算法需要指定蚂蚁的行动路线，而问题三则不需要限定蚂蚁的行动路线。问题三的结果如下所示：

表 6.3.1 2023 年 1 月删除 DC9 后新增的路线数据

场地 1	场地 2	添加时间	流量
3	79	2023/1/1	53
3	79	2023/1/2	114
3	79	2023/1/3	263
3	79	2023/1/4	640
3	79	2023/1/5	1092
3	79	2023/1/6	929
3	79	2023/1/7	683
3	79	2023/1/8	608
3	79	2023/1/9	516
3	79	2023/1/10	488
3	79	2023/1/11	507
3	79	2023/1/12	563
3	79	2023/1/13	621
3	79	2023/1/14	715
3	79	2023/1/15	836
3	79	2023/1/16	992
3	79	2023/1/17	1195
3	79	2023/1/18	1444
3	79	2023/1/19	1753
3	79	2023/1/20	2142
3	79	2023/1/21	2634
3	79	2023/1/22	3253

由表可知，关停 DC9 后仅新增 DC3→DC79 路线，由于物流场所 DC79 只与 DC9 来往，关停 DC9 后添加 DC3 作为中转点，新增路线 1 月的平均送货量为 1204.86

关停 DC9 后每条路线的最大送货量/月平均送货量、超出/剩余货量及负荷率情况如下表所示。

表 6.3.2 正常流转的货量以及工作负荷

场地 1	场地 2	最大送货量	1 月平均送货量	超出/剩余货量	负荷率
69	5	1870	1865	0	100
69	8	28132	28127	0	100
69	14	4479	4474	0	100
69	62	5174	5169	0	100
69	10	10561	10556	540.2044	94.264
60	5	5546	5541	1128.195	78.854
23	32	604	599	251.8933	53.854
64	8	1696	1691	774.2042	52.574
60	8	21799	21794	10205.2	51.604
60	10	7433	7428	3550.415	50.874
27	10	16133	16128	7827.474	50.664
8	14	32291	32286	15697.52	50.874
52	8	3998	3993	1977.441	49.884
26	64	11493	11488	5836.855	48.594
4	23	3519	3514	1803.861	48.074
70	4	385	380	196.0546	47.864
38	64	14273	14268	7562.925	46.404
30	5	3986	3981	2110.57	46.394
27	8	57062	57057	30667.09	45.654
47	14	64007	64002	34613.2	45.324

由表可知，DC69→DC5、DC69→DC8、DC69→DC14、DC69→DC62 的负荷率为 100%，其余路线的负荷率均未达到 100%，并且平均路线负荷率为 63.07%，所有路线的平均货物剩余量为 1237.155，说明删除 DC9 并且新增 DC3→DC79 路线后对整体配送结果没有影响，无超出负荷的路线，满足实际要求。

6.4 问题四模型的建立与求解

6.4.1 网络的不同物流场地及线路的重要性评价

由问题二可以得到，物流场所与物流路线之间的相对关系。要评价网络中物流点和线路的重要性，可以使用网络分析库来计算各种度量。

对于节点，可以用度中心性度量、紧密中心性度量、介数中心性度量和特征向量中心性度量四种方法进行度量。其中度中心性度量即节点的度数除以所有节点的度数之和；紧密中心性度量即节点到其他节点的平均距离的倒数，其中距离可以是路径长度或加权路径长度；介数中心性度量即节点在网络中连接其他节点之间的最短路径数量占有所有最短路径数量的比例。

对于线路，可以计算边介数中心性（edge betweenness centrality）进行度量，即边在网络中连接其他节点之间的最短路径数量占有所有最短路径数量的比例。根据各物流场所、物流路线之间的物流能力配送系数级功能坐标点。

通过以上五种度量方法得到的结果并排序。下列展示部分重要性指数较高的前几位物流点及物流路线。

Node degree:	Node betweenness:	Node eigenvector:
10: 0.9125000000000001	3: 0.1859410890621334	14: 0.28229649340174184
14: 0.9125000000000001	57: 0.16265822784810127	17: 0.25967635348396784
5: 0.8500000000000001	5: 0.14573564401491618	36: 0.25305835727933484
8: 0.8250000000000001	10: 0.14353627692630858	8: 0.24104459312497128
4: 0.8	9: 0.1334382578567705	51: 0.2385755294849996
3: 0.775	39: 0.12563291139240507	22: 0.22314842046649888
62: 0.7375	35: 0.1085245982041868	10: 0.21708363280208515
9: 0.7250000000000001	8: 0.1061780207310271	20: 0.2070536603428943
35: 0.6875	14: 0.08541918831871365	12: 0.1973973313501244
40: 0.6000000000000001		19: 0.19570878408821743

Node closeness:	Edge betweenness:
3: 0.1859410890621334	('9', '39'): 0.08251028806584365
57: 0.16265822784810127	('35', '57'): 0.07789155888692925
5: 0.14573564401491618	('3', '76'): 0.07261316872427985
10: 0.14353627692630858	('3', '23'): 0.06574074074074074
9: 0.1334382578567705	('57', '70'): 0.05648062414266118
39: 0.12563291139240507	('23', '54'): 0.04876543209876543
35: 0.1085245982041868	('8', '39'): 0.04856481481481482
8: 0.1061780207310271	('30', '35'): 0.046296296296296294
14: 0.08541918831871365	('5', '39'): 0.04342078189300411
	('10', '39'): 0.039639917695473245

图 6.4.1 五种度量方法部分重要性指数较高的物流点及物流路线结果图

6.4.2 新物流网络的建立

根据上节结果可以得到，DC14、DC10、DC5、DC8、DC3 在整个物流网络中重要性较高，即对物流网络的影响比较大。因此，如果考虑新增物流场地及线路可以考虑在与这五个物流场地较近的地方新增物流场地，然后新增连接原有场地的路线，以解决网络超负荷运转问题。由于新增场地与原有网络场地较接近，为缓解原有场地网络负荷问题，新增路线前所有场地处理能力之和等于新增场地后所有场地处理能力之和，因此新增场地的处理能力应该大于与其距离较近的原有场地处理能力的 1/2 倍而小于原有场地处理能力。

基于以上分析，根据以上五个物流点的物流点处理能力以及物流线路的运输能力确定一个新物流运输点 82，坐标如下。

```

hess_inv: array([[1.01431596e-07, 5.88305990e-09],
                 [5.88305990e-09, 1.79350141e-09]])
jac: array([1.06561739, 2.04221044])
message: 'Desired error not necessarily achieved due to precision loss.'
nfev: 258
nit: 20
njev: 82
status: 2
success: False
x: array([-0.05655923, -0.06336183])

```

图 6.4.2 DC82 位置确定结果

根据预测中对其原有路线的运输量动态调整发现，当新增路线 DC14→DC82, DC8→DC82, DC82→DC8, DC82→DC10, DC82→DC5, DC82→DC14 时在新增路线少的前提下提高路线运输能力，其运输能力参数分别为 93691, 16590, 68350, 24860, 11149, 5982。通过分析，在新增了物流点后，五个原物流点的物流处理饱和率分别降低了 24.6%，18.5%，21.3%，19.7%，13.3%。

6.4.3 鲁棒性检验

物流网络的鲁棒性是指在站点遭受突发事件时仍能保持网络正常有序运行的能力。由于上述不同的中心性计算方法针对不同方面对物流点进行了重要性排序，因此本文采用对四种中心性计算结果的平均值作为本小节讨论的重要值。以下是物流网络鲁棒性的评价指标。

网络稳定性是指网络遭遇外部冲击时网络仍然保持正常运行的能力。在网络遭遇攻击时，如果物流点路线运货频率下降趋势比较大，网络稳定性就比较差，因此将路线货量作为稳定性的度量指标。基于加权物流网络构建物流网络的稳定性指标。对于物流网络来说，如果物流路线货量挤压损失越小，那么物流路线网络的稳定性越高，就具有更高的抵御外部冲击的能力。

除网络稳定性指标以外，引入网络效率指标来表示物流路线网络的连通性。网络效率性指标体现网络运行的效率，是衡量网络运输能力的关键指标。在物流网络中如果效率性越好，说明物流路线网络传输速率越快，网络连通性越强^[6]。当站点之间的距离越短时，则站点的效率越高，可用于衡量物流网络的运行效率，整个物流网络的网络效率 E 定义为网络中站点之间所有距离的倒数平均值，计算公式如下，其中 N 为网络站点总数， d_{ij} 为站点 v_i 与 v_j 之间的距离，若 v_i 与 v_j 之间不连通，则有 $d_{ij} \rightarrow +\infty, 1/d_{ij} \rightarrow 0$ 。

$$E = \frac{2}{N(N-1)} \sum_{i,j=1(i \neq j)}^N \frac{1}{d_{ij}} \quad (6.4.1)$$

综合以上两个评价指标引入到网络鲁棒性的计算中，公式如下：

$$\eta = \left(1 - e^{-(\varepsilon_1 \eta_{Sj} + \varepsilon_2 \eta_{Ej})}\right) * \frac{d_j}{d_0} \quad (6.4.2)$$

其中 η 是网络的鲁棒性， η_{Sj} 是删除节点 j 后网络稳定值， η_{Ej} 是删除节点 j 后的全局网络效率值； d_j 是删除节点 j 后的剩余节点度之和； d_0 是原始网络中的节点度之和； ε_1 、

ε_2 是两个调整参数。本文令 $\varepsilon_1 = \varepsilon_2 = 0.5$ 。

采用 python 进行编程，对优化前后的物流点物流路线网络进行攻击，采用随机攻击和蓄意攻击两种攻击方法对比分析其物流网络的鲁棒性。随机攻击对应自然灾害、信号故障、设备故障等随机故障，蓄意攻击则对应促销节日活动等类似行为，主要针对网络中的重要站点进行攻击。随机攻击是利用 python 编程生成随机序列对已编号的网络站点进行移除。蓄意攻击又分为度值攻击、介数攻击及重要度攻击，分别按照节点度值、介数值和前文确定的节点重要度降序排序，然后在网络依次移除相关站点。

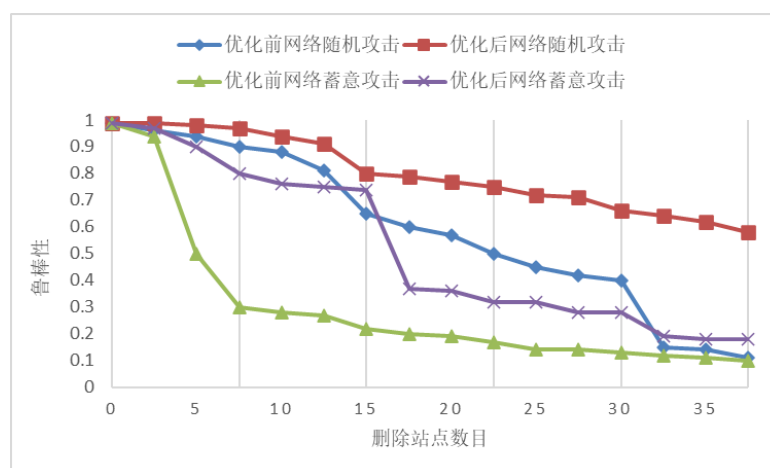


图 6.4.3 不同网络与不同攻击下鲁棒性系数对比图

如图为优化前后物流网络面对不同攻击策略时网络效率的变化情况。从中可以看出，优化后随机攻击下，鲁棒性系数下降缓慢，这说明小范围内随机故障对网络整体运行影响不大，但随着故障规模的扩大也会对网络造成较大影响，当移除节点数达到 35 个时，鲁棒性系数仅为 0.6，而优化前网络在随机攻击情况下，鲁棒性系数下降与优化后趋势相差不大，但当移除节点数达到 30 个时发生突降，这可以解释为随机攻击移除了网络中的重要站点，从而引发多轮负载重新分配过程，造成多个站点失效，产生较大影响，导致鲁棒性系数骤降。网络在面对蓄意攻击时，鲁棒性系数快速下降，在优化后的情况下，当移除节点数为 18 个时，重要度攻击下的鲁棒性系数下降到 0.36，这说明攻击的这部分站点在网络中尤为关键，发挥着关键作用，且重要程度高的节点失效之后将会给整个网络带来更加严重的后果。在优化前情况下，攻击重要度前 8 的站点造成鲁棒性系数下降为 0.29，当攻击站点数达到 35 个时，鲁棒性系数下降至 0.1，网络受到毁灭性打击，无法正常运行。通过对比优化前后网络的鲁棒性可以发现优化后的网络最终的鲁棒性要高于优化前。

七、模型的评价

模型的优点：

1.LSTM-ARIMA 可以较好地捕捉到时间序列之间的相关性，相对单纯的 ARIMA 模型可以捕捉时间序列的非线性关系，可作为复杂的非线性单元用于构造更大型深度神经

网络，能够有效减缓长序列问题中可能出现的梯度消失或爆炸，在更长的序列问题上表现优于传统 RNN。

2.多维尺度变换 MDS 算法是一种数据降维和可视化方法，能够将高维度数据转换到低维度空间，在保持数据点间距离关系的同时直观地观察和分析数据。

3.多目标规划的蚁群算法采用正反馈机制，使得搜索过程不断收敛，最终逼近最优路线；每个个体可以通过释放信息素来改变周围的环境，且每个个体能够感知周围环境的实时变化，个体间通过环境进行间接地通讯；搜索过程采用分布式计算方式并行计算，提高了算法的计算能力和运行效率。启发式的概率搜索方式不容易陷入局部最优，易于寻找到全局最优线路。

模型的缺点：

1.LSTM-ARIMA 组合模型并行处理上存在劣势,只能从前到后,与一些最新的网络相对效果一般，对于更长的时间序列预测效果不够好。

2.在本文中数据量较大时，运算时间较长，功能坐标间距离为相互路线配送能力系数的均值，不能较好地转化路线有向性数据。此外，由 MDS 算法得到的功能坐标认为各个维度的地位相同，无法区分不同维度的重要性。

3.由于蚁群算法对路径规划多样性要求较高，如果多样性过剩，系统过于活跃会导致过多的随机运动，陷入混沌状态。多样性不够，正反馈过强，会导致僵化，当环境变化时蚁群不能相应调整。

参考文献

- [1]徐虎博,史东辉.基于 ARIMA 和 LSTM 模型的建筑安全事故预测[J].软件工程,2023,26(03):9-14.
- [2]杨彬.基于 MDS 的无人机群协同定位算法研究[D].西安电子科技大学,2021.
- [3]王雪凤,陈昊.基于物流配送的车辆路径优化问题综述[J].辽宁工业大学学报(自然科学版),2022,42(06):386-392.
- [4]金梦圆,王勇畅,刘敬琪.基于蚁群算法的物流多任务分配中路径规划研究[J].计算机时代,2023(03):40-43.
- [5]周琪,张谦述,张耀进.改进蚁群算法在多目标下路径长度的研究[J].邵阳学院学报(自然科学版),2023,20(01):32-38.
- [6]王世瑞,吴薇薇.基于航线网络结构特征的鲁棒性研究[J].哈尔滨商业大学学报(自然科学版),2022,38(06):714-722.

九、附件

- 附录 1：支撑材料文件列表
附录 2：补充表格、图片或者公式推导
附录 3：代码展示

附录 1
支撑材料文件列表
问题一相关代码 ARIMA 预测代码.ipynb LSTM 预测代码.m ARIMA-LSTM 加权聚合算法.py CRITIC 算法代码.py
问题二相关代码 路径匹配载荷代码.py MDS 算法代码 1.py MDS 算法代码 2.py 蚁群算法代码.py
问题三相关代码 蚁群算法代码.py
问题四相关代码 中心性度量代码.py

附录 2 ： 补充的表格、图片或者公式推导

时间序列预测表									
时间	ARIMA 时间序列预测值			LSTM 时间序列预测值			ARIMA-LSTM 时间序列预测值		
	路线 14→	路线 20→	路线 25→	路线 14→	路线 20→	路线 25→	路线 14→	路线 20→	路线 25→
	10 预测结果	35 预测结果	62 预测结果	10 预测结果	35 预测结果	62 预测结果	10 预测结果	35 预测结果	62 预测结果
1	25898	204	9005	20385	0	2021	23306.5	76	5476.5
2	25933	214	9269	14175	0	1900	20413.5	79	5536.5
3	25968	227	9305	19207	0	2493	23133.5	83	5844.5
4	26003	228	9320	24249	0	3093	25831.5	83.5	6145.5
5	26038	230	9333	27721	0	3681	27722	83.5	6439.5
6	26072	230	9346	30659	0	4185	29325	84	6691.5
7	26107	231	9359	33001	4	4652	30612.5	86	6925
8	26142	231	9372	34943	10	5048	31685	89	7123
9	26177	231	9385	36514	16	5388	32558.5	92	7293
10	26212	232	9398	37803	21	5683	33280	94.5	7440.5
11	26247	232	9410	38907	26	5946	33898.5	97	7572
12	26281	232	9423	39888	30	6188	34447	99	7693
13	26316	232	9436	40777	35	6413	34942	101.5	7805.5
14	26351	233	9449	41592	38	6623	35393.5	103	7910.5

15	26386	233	9462	42348	42	6821	35809.5	105	8009.5
16	26421	233	9474	43046	45	7007	36192	106.5	8102.5
17	26456	234	9487	43704	48	7184	36549.5	108	8191
18	26490	234	9500	44337	51	7356	36891.5	109.5	8277
19	26525	234	9513	44951	54	7524	37220	111	8361
20	26560	235	9526	45542	56	7688	37534.5	112	8443
21	26595	235	9539	46111	59	7847	37835.5	113.5	8522.5
22	26630	235	9551	46682	62	8005	38135.5	115	8601.5
23	26665	236	9564	47244	65	8162	38429	116.5	8680
24	26699	236	9577	47792	67	8316	38714	117.5	8757
25	26734	236	9590	48321	70	8466	38988	119	8832
26	26769	237	9603	48824	73	8609	39247.5	120.5	8903.5
27	26804	237	9615	49299	75	8743	39492	121.5	8970.5
28	26839	237	9628	49737	78	8867	39717.5	123	9032.5
29	26874	238	9641	50132	80	8979	39920.5	124	9088.5
30	26908	238	9654	50489	82	9079	40103.5	125	9138.5
31	27397	238	9667	50815	83	9167	40270.5	125.5	9182.5

场地功能坐标表

场地	x	y
DC1	-0.02901	0.835956
DC10	-0.05792	-0.04471
DC11	-0.52488	0.594656
DC12	-0.15928	0.101051
DC13	0.372052	-0.46892
DC14	-0.05656	-0.06336
DC15	0.251743	-0.42864
DC16	0.317388	-0.70141
DC17	0.281158	-0.22897
DC18	-0.71438	0.332493
DC19	0.195236	-0.12415
DC2	0.268635	0.634535
DC20	-0.02918	-0.27105
DC21	-0.22107	-0.25179
DC22	-0.24935	-0.1268
DC23	-0.20646	0.106545
DC24	0.68112	-0.25968
DC25	-0.46704	-0.1259
DC26	-0.39575	-0.47183
DC27	-0.25571	-0.53074
DC28	-0.41046	0.18472
DC29	0.604278	0.047403
DC3	-0.06026	-0.01221
DC30	-0.07181	0.180427
DC31	-0.58068	0.115029

DC32	-0.30914	-0.00412
DC33	0.066296	-0.57328
DC34	-0.12176	0.021509
DC35	-0.09269	0.003319

部分物流路线最大载荷、平均载荷及配送能力结果

场地 1	场地 2	路线载荷	平均载荷	配送能力
DC1	DC8	3	3	3
DC10	DC12	993	320.3886	178338
DC10	DC13	6	3.190476	178338
DC10	DC14	31846	7879.552	178338
DC10	DC17	5	2	178338
DC10	DC19	14	6.6	178338
DC10	DC20	11	4.25	178338
DC10	DC21	29	6.679012	178338
DC10	DC22	27	5.955556	178338
DC10	DC23	991	72.59259	178338
DC10	DC25	4	3	178338
DC10	DC27	6	6	178338
DC10	DC28	8	2.666667	178338
DC10	DC3	27671	5572.365	178338
DC10	DC30	50	4.895522	178338
DC10	DC32	206	20.84848	178338
DC10	DC33	118	5.920635	178338
DC10	DC34	200	12.43103	178338
DC10	DC35	1914	347.1043	178338
DC10	DC36	55	7.020408	178338

附录 3：相关代码展示

问题一 LSTM 算法相关代码：

```
res=readmatrix("OriginDataResult_3_lstm.xlsx");
res(isnan(res)) = 0;
%% 划分训练集和测试集
%temp = randperm(730);
P_train = res(1: 700, :);
T_train = res(1: 700, :);
M = size(P_train, 2);
P_test = res(701: end, :);
T_test = res(701: end, :);
N = size(P_test, 2);
%% 数据归一化
[P_train, ps_input] = mapminmax(P_train, 0, 1);
P_test = mapminmax('apply', P_test, ps_input);
```

```

[t_train, ps_output] = mapminmax(T_train, 0, 1);
t_test = mapminmax('apply', T_test, ps_output);
%% 数据平铺
P_train = double(reshape(P_train, 1049, 1, 1, M));
P_test = double(reshape(P_test, 1049, 1, 1, N));
t_train = t_train';
t_test = t_test';
%% 数据格式转换
p_train = cell(M, 1);
p_test = cell(N, 1);
for i = 1 : M
    p_train{i, 1} = P_train(:, :, 1, i);
end
for i = 1 : N
    p_test{i, 1} = P_test(:, :, 1, i);
end
%% 定义权重向量
decay_rate = 0.9; % 指数衰减率
% W = decay_rate.^(0:M-1); W = W / sum(W); % 归一化
%% 创建模型
layers = [
    sequenceInputLayer(1049) % 建立输入层
    lstmLayer(512, 'OutputMode', 'sequence', ...
    InputWeights=randn(512*4,1049), ...
    GateActivationFunction="hard-sigmoid") % LSTM 层
    reluLayer % Relu 激活层
    dropoutLayer(0.5) % 添加 Dropout 层，丢弃概率为 0.2
    %
    % lstmLayer(512,'OutputMode', 'last', ...
    %
    Bias=[zeros(1,512),ones(1,512),zeros(1,512),ones(1,512)],GateActivationFunction="sigmoid")
    % % LSTM 层 sigmoidLayer % Relu 激活层
    % dropoutLayer(0.2) % 添加 Dropout 层，丢弃概率为 0.2
    %
    % lstmLayer(512, 'OutputMode', 'last', ...
    % Bias=[ones(1,512),zeros(1,512),ones(1,512),zeros(1,512)],
    BiasLearnRateFactor=N) % LSTM 层
    % tanhLayer % Relu 激活层 dropoutLayer(0.2)
    % %添加 Dropout 层，丢弃概率为 0.2
    lstmLayer(128, 'OutputMode', 'last', ...
    InputWeights=[], CellState=randn(128,1),...
    GateActivationFunction='sigmoid') % LSTM 层
    sigmoidLayer % Relu 激活层
    dropoutLayer(0.1) % 添加 Dropout 层，丢弃概率为 0.2
    %
    lstmLayer(64,'OutputMode', 'sequence') % LSTM 层 sigmoidLayer %

```

```

% Relu 激活层 dropoutLayer(0.01) % 添加 Dropout 层, 丢弃概率为 0.2
lstmLayer(32,'OutputMode','sequence')
reluLayer
dropoutLayer(0.1)
lstmLayer(32,'OutputMode','sequence')
reluLayer
dropoutLayer(0.1)
lstmLayer(32,'OutputMode','sequence') % LSTM 层
reluLayer % Relu 激活层
dropoutLayer(0.01) % 添加 Dropout 层, 丢弃概率为 0.2

lstmLayer(16,'OutputMode','sequence') % LSTM 层
reluLayer % Relu 激活层
dropoutLayer(0.01) % 添加 Dropout 层, 丢弃概率为 0.2
fullyConnectedLayer(1049) % 全连接层
regressionLayer]; % 回归层

%% 参数设置
options = trainingOptions('adam', ... % Adam 梯度下降算法
    'MiniBatchSize', 1024, ... % 批大小
    'MaxEpochs', 1200*5, ... % 最大迭代次数
    'ValidationFrequency',50,...
    'InitialLearnRate', 0.001, ... % 初始学习率为
    'LearnRateSchedule', 'piecewise', ... % 学习率下降
    'LearnRateDropFactor', 0.2, ... % 学习率下降因子
    'LearnRateDropPeriod', 5000, ... % 经过 800 次训练后 学习率为 0.01 * 0.5
    'Shuffle', 'every-epoch', ...
    'ExecutionEnvironment','gpu',...% 每次训练打乱数据集
    'SquaredGradientDecayFactor',0.7,...
    'Plots', 'training-progress', ... % 画出曲线
    'Verbose', false);

%% 训练模型
net = trainNetwork(p_train, t_train, layers, options);

%% 仿真预测
t_sim1 = predict(net, p_train);
t_sim2 = predict(net, p_test);

%% 查看网络结构
analyzeNetwork(net)

%% 数据反归一化
T_sim1 = mapminmax('reverse', t_sim1, ps_output);
T_sim2 = mapminmax('reverse', t_sim2, ps_output);
for i = 1:2
    %% 均方根误差
    error1(i,:) = sqrt(sum((T_sim1(:,i) - T_train(i,:)).^2) ./ M);
    error2(i,:) = sqrt(sum((T_sim2(:,i) - T_test(i,:)).^2) ./ N);
    %% 绘图
    figure

```

```

subplot(2, 1, 1)
plot(1: M, T_train(i,:), 'r-*', 1: M, T_sim1(:,i), 'b-o', 'LineWidth', 1)
legend('True value', 'Predicted value')
xlabel('Predict the sample')
ylabel('Predict the outcome')
string = {'Comparison of training set prediction results'; ['RMSE=' num2str(error1(i,:))]};
title(string)
xlim([1, M])
grid
subplot(2, 1, 2)
plot(1: N, T_test(i,:), 'r-*', 1: N, T_sim2(:,i), 'b-o', 'LineWidth', 1)
legend('True value', 'Predicted value')
xlabel('Predict the sample')
ylabel('Predict the outcome')
string = {'Comparison of test set prediction results'; ['RMSE=' num2str(error2(i,:))]};
title(string)
xlim([1, N])
grid
%% 分割线
disp('*****')
disp(['下列是输出', num2str(i)])
disp('*****')
%% 相关指标计算
% R2
R1(i,:) = 1 - norm(T_train(i,:) - T_sim1(:,i))^2 / norm(T_train(i,:) - mean(T_train(i,:)))^2;
R2(i,:) = 1 - norm(T_test(i,:) - T_sim2(:,i))^2 / norm(T_test(i,:) - mean(T_test(i,:)))^2;
disp(['The R2 of the training set data is: ', num2str(R1(i,:))])
disp(['The R2 of the test set data is: ', num2str(R2(i,:))])
% MAE
mae1(i,:) = sum(abs(T_sim1(:,i) - T_train(i,:))) ./ M ;
mae2(i,:) = sum(abs(T_sim2(:,i) - T_test(i,:))) ./ N ;
disp(['The MAE of the training set data is: ', num2str(mae1(i,:))])
disp(['The MAE of the test set data is: ', num2str(mae2(i,:))])
% MBE
mbe1(i,:) = sum(T_sim1(:,i) - T_train(i,:)) ./ M ;
mbe2(i,:) = sum(T_sim2(:,i) - T_test(i,:)) ./ N ;
disp(['The MBE of the training set data is: ', num2str(mbe1(i,:))])
disp(['The MBE of the test set data is: ', num2str(mbe2(i,:))])
end
%% 进行向后预测
% 定义需要预测的时间步数
futureTimeSteps = 31;
% 取最后一段测试数据作为输入
lastSequence = p_test{end};
lastSequence=lastSequence';
% 初始化预测结果矩阵

```

```

predictedResults = zeros( futureTimeSteps,1049);
predictedResults(1, :) =lastSequence;
% net = predictAndUpdateState(net,p_test);
% 预测未来的数据
for i = 1:futureTimeSteps
    lastSequenceT=predictedResults(i,:);
    lastSequenceT=lastSequenceT';
    % 使用模型进行预测
    predictedValue = predict(net, lastSequenceT);
    % 将预测值添加到结果中
    predictedResults(i+1,:) = predictedValue;
end
% 将预测结果反归一化
predictedResults = mapminmax('reverse', predictedResults', ps_output)';
predictedResults(predictedResults<1)=0;
predictedResults=roundn(predictedResults,0);
% writematrix(predictedResults, 'predicted_result.xlsx');

```

问题二 CRITIC 算法相关代码：

```

import pandas as pd
import numpy as np

# 读取 Excel 文件，获取 Sheet1 和 Sheet2 的数据
excel_file = pd.ExcelFile("D:\\Desktop\\example.xlsx")
data_sheet1 = excel_file.parse("Sheet1")

# 将第一列和第二列的数据存储到 data1 和 data2 中
data1 = data_sheet1.iloc[:, 0].tolist()
data2 = data_sheet1.iloc[:, 1].tolist()

```

```

# 计算相关系数矩阵
corr_matrix = np.corrcoef([data1,data2])
print("相关系数: \n", corr_matrix)
# 计算每个属性的权重
weights = []
for i in range(len(corr_matrix)):
    weights.append(sum(corr_matrix[i]) / len(corr_matrix))

# 归一化权重
weights = weights / sum(weights)
print("属性权重: ", weights)

```

问题四重要性度量相关代码:

```

import networkx as nx

# 构造无向图
G = nx.Graph()

# 添加节点及权重（运货量）
node_weights = {'1': 3, '2': 250, '3': 2611, '4': 27386, '5': 16980, '6': 1832, '7': 14128, '8': 168725, '9': 72608, '10': 178338, '11': 14036, '12': 551, '13': 664, '4': 66634, '15': 109833, '16': 31516, '17': 437920, '18': 97230, '19': 320853, '20': 186445, '21': 182916, '22': 402134, '23': 286082, '24': 36, '25': 159822, '26': 49816, '27': 95355, '28': 84320, '29': 1, '30': 64482, '31': 22266, '32': 1323, '33': 3052, '34': 8196, '35': 3185, '36': 369682, '37': 8433, '38': 156577, '39': 15, '40': 33021, '41': 35763, '42': 35227, '43': 85702, '44': 257960, '45': 36681, '46': 100327, '47': 168273, '48': 93463, '49': 39976, '50': 65406, '51': 400791, '52': 5209, '53': 26382, '54': 53, '55': 3406, '56': 2564, '57': 425, '58': 76509, '59': 31036, '60': 47685, '61': 369, '62': 146907, '63': 30, '64': 3347, '65': 149, '66': 587, '67': 3792, '68': 29457, '69': 50191, '70': 1241, '71': 22358, '72': 10016, '73': 282924, '74': 5331, '75': 2, '76': 24, '77': 2}
for node, weight in node_weights.items():
    G.add_node(node, weight=weight)

# 添加边及边权重
import csv

edge_weights = {}

with open('D:\\Desktop\\重要性测试.csv', 'r') as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        edge_weights[(row[0], row[1])] = int(row[2])
for edge, weight in edge_weights.items():
    G.add_edge(*edge, weight=weight)

```



```
# 计算各种中心性指标
node_degree = nx.degree centrality(G)#节点度中心性
node_closeness = nx.closeness centrality(G)#紧密中心性
node_betweenness = nx.betweenness centrality(G, weight='weight')#介数中心性
node_eigenvector = nx.eigenvector centrality(G, weight='weight')#特征向量中心性
edge_betweenness = nx.edge_betweenness centrality(G, weight='weight')

print("Node degree:")
for node, degree in sorted(node_degree.items(), key=lambda x: x[1], reverse=True):
    print(f'{node}: {degree}')

print("Node closeness:")
for node, closeness in sorted(node_closeness.items(), key=lambda x: x[1], reverse=True):
    print(f'{node}: {closeness}')

print("Node betweenness:")
for node, betweenness in sorted(node_betweenness.items(), key=lambda x: x[1], reverse=True):
    print(f'{node}: {betweenness}')

print("Node eigenvector:")
for node, eigenvector in sorted(node_eigenvector.items(), key=lambda x: x[1], reverse=True):
    print(f'{node}: {eigenvector}')

print("\nEdge betweenness:")
for edge, betweenness in sorted(edge_betweenness.items(), key=lambda x: x[1], reverse=True):
    print(f'{edge}: {betweenness}')
```